UNIVERSITÉ DE SHERBROOKE

Faculté de Génie

Département de génie chimique

MÉMOIRE DE MAÎTRISE

ÉTUDE DE DONNÉES SPATIO-TEMPORELLES POUR L'ANALYSE DU CONTRÔLE ENVIRONNEMENTAL EN MILIEU INDUSTRIEL PHARMACEUTIQUE

Alexandre Vielfaure

Juin 2020

MEMBRES DU JURY

Ryan Gosselin

Directeur de recherche

Marc-Antoine Lauzon

Évaluateur Interne

Antoine Cournoyer

Évaluateur Externe

Nicolas Abatzoglou

Rapporteur

RÉSUMÉ

Une quantité importante de procédés industriels sont aujourd'hui monitorés à l'aide de capteurs et d'analyses afin d'avoir de l'information sur l'état des opérations et faciliter la réponse à d'éventuelles déviations. Bien que l'analyse de ces données soit une pratique de plus en plus courante dans l'industrie, l'étude de données spatio-temporelles (avec de l'information répartie à la fois dans l'espace et dans le temps) comportant un important niveau de bruit représente encore un défi.

Dans ce projet, des données spatio-temporelles historiques d'un programme de contrôle environnemental en milieu pharmaceutique ont été récoltées pour mieux comprendre les dynamiques de contamination entre les zones aseptiques de productions. Un défi majeur pour l'analyse de ce genre de données est la présence importante de bruit en raison de la rareté des résultats non-nuls et de l'incertitude reliée à la détection de microorganismes. Les objectifs principaux du projet étaient donc l'étude des données spatio-temporelles et le développement d'outils algorithmiques permettant de faciliter l'interprétation des résultats.

Dans un premier temps, un nouvel indice de similarité qui utilise une combinaison de la corrélation de Pearson et le « dynamic time warping » a été développé. Cet indice, employé pour la recherche de similarité entre variables, permet de mieux traduire les dynamiques de contamination dans les données de contrôle environnemental. En combinant les résultats des recherches de similarité avec des outils de visualisation, les patrons de contamination entre les différentes zones de productions ont pu facilement être mis en évidence.

Dans un second temps, une nouvelle approche multivariée pour l'étude de données spatio-temporelles fortement bruitées à l'aide de l'algorithme MCR-LLM a aussi été explorée. Cette méthode, précédemment développée pour l'analyse de données spectroscopiques, a permis d'extraire des composants représentant les différents patrons de contamination dans les données de contrôle environnemental. L'application de cette méthode a grandement facilité l'étude des données en mettant en évidence les principales dynamiques spatiales présentes et en simplifiant la visualisation des variations temporelles. Les approches présentées peuvent aussi être utilisées sur d'autres jeux de données avec des caractéristiques spatio-temporelles similaires.

Mots clés : Données spatio-temporelles, Analyse Multivariée, Multivariate Curve Resolution, Contrôle Environnemental, Pharmaceutique, Dynamic Time Warping

REMERCIEMENTS

J’aimerais tout d’abord remercier Ryan pour son aide avec mes questions plus techniques et mathématiques lors de ma maîtrise. Tes conseils et commentaires m’ont beaucoup aidé à mieux comprendre les différents techniques et algorithmes que j’ai eu à utiliser.

Merci aussi à Antoine qui m’a énormément appris sur le plan professionnel. Tout le temps que tu as pris pour me guider et de répondre à mes questions est extrêmement apprécié.

Je veux aussi prendre le temps de dire un gros merci à mes collègues chez Pfizer et plus particulièrement Charlotte, Barbara et Azher. Merci pour tous les conseils, encouragements et bons moments passés ensemble.

Merci à mes parents qui m’ont soutenu et encouragé pendant toutes mes études.

Je ne peux pas terminer sans remercier ma copine Farah qui a été ma partenaire et ma complice depuis le tout début.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

1. INTRODUCTION

1.1. Contexte et problématique

Les industries pharmaceutiques et alimentaires accordent de nos jours de plus en plus d'importance au contrôle qualité afin d'assurer que les produits qui sortent des usines respectent les normes en places et sont sécuritaires pour la consommation. Pour ce faire, celles-ci mettent en places des mesures importantes afin d'assurer un contrôle de leurs produits aux différentes étapes de la production tout comme un contrôle en soi des environnements de production.

Les agences gouvernementales telles que la FDA (Food and Drug Administration), l'EMA (European Medecines Agency) ou l'EFSA (European Food Safety Authority) ont aussi mis en place des exigences strictes pour assurer le respect des normes et des bonnes pratiques de production. Bien que les tests et analyses sur les produits soient à la base du contrôle qualité, le suivi et l'analyse des zones de productions sont aussi une partie essentielle du processus de fabrication de médicaments. Dépendamment de l'industrie et du type de produit, l'ampleur et l'importance du contrôle environnemental des zones de fabrication vont cependant varier.

Dans le cas de l'industrie pharmaceutique, une importance particulière est accordée à cet aspect de la production puisque des produits contaminés sont à risque de créer de graves problèmes pour le consommateur. Les acteurs de l'industrie se doivent de respecter les bonnes pratiques de fabrication (cGMP) décrites dans les documents de la FDA (21CFR§211) ainsi que dans la Pharmacopée américaine (USP <797>). Ces pratiques présentent principalement l'utilisation de programmes environnementaux qui définissent des mesures préventives (design adéquat du bâtiment, maintenance, contrôle des procédés, plans de nettoyage, etc.) et des exigences pour limiter la contamination des zones critiques de productions.

Un programme environnemental adéquat touche plusieurs secteurs d'activités différents. Il permet à la fois de savoir si l'usine est dans un état de contrôle par rapport à la contamination des zones de productions, mais aussi d'apporter de l'information utile pour l'identification des sources de contamination et les actions éventuelles à mettre en place pour les contrer. Une des dimensions les plus importantes du contrôle environnemental est la récolte, la gestion et l'analyse des données générées. En effet, une quantité

importante d'analyses et d'échantillonnages sont réalisés quotidiennement dans les zones contrôlées afin de détecter les éventuelles déviations. Ceci génère donc énormément de données à la fois spatiales puisqu'on teste dans plusieurs zones différentes et temporelles en raison de la fréquence généralement élevée des analyses.

Les outils utilisés pour analyser les données environnementales en milieu pharmaceutique sont cependant très limités et se basent généralement uniquement sur des seuils et des tableaux simples présentant les données brutes. Aucune méthode se basant sur une analyse multivariée et prenant en compte les aspects spatio-temporels des données n'a été présentée.

De façon générale, l'utilisation d'analyses peu adaptées aux données spatio-temporelles mène souvent à une mauvaise compréhension de celles-ci. Dans le cas des données de contrôle environnemental, cela peut engendrer une mauvaise utilisation des ressources comme des nettoyages trop fréquents, une plus grande quantité d'analyses réalisées et une mauvaise utilisation de la main d'œuvre par exemple. De plus, une mauvaise compréhension du processus rend l'identification des sources de contamination plus difficile et donc augmente le risque de contamination des produits.

L'analyse et l'interprétation de ces données relèvent donc de l'étude spatio-temporelle d'une problématique industrielle dans le but d'améliorer la compréhension et le contrôle d'un procédé. Il est donc nécessaire d'utiliser des approches adaptées qui prennent en compte les caractéristiques des données pour les analyser en profondeur et faire ressortir l'information pertinente. Ce type de données et d'analyses présentent cependant des défis supplémentaires en raison des dépendances spatiales et temporelles des observations, des incertitudes et du bruit important. Il n'existe aujourd'hui aucun outil ou méthodologie permettant de faire face à une problématique avec les caractéristiques suivantes :

- Processus complexe qui dépend de nombreux paramètres
- Données spatio-temporelles comportant différentes résolutions (spatiale et temporelle)
- Relations et dépendances complexes entre les variables
- Influence de facteurs difficilement mesurables qui peuvent impacter la modélisation
- Présence de bruit élevé dans les données
- Événements peu fréquents (rares) en raison de la nature des données

Cette situation n'est donc pas optimale et un changement dans l'approche des problèmes avec des données spatio-temporelles en milieu industriel est désiré. Bien que les données du projet soient issues du contrôle environnemental en zone de production pharmaceutique, la problématique est très large et se prête à tout type de procédé ayant des données spatio-temporelles. De plus, avec l'augmentation importante de la quantité d'information collectée dans plusieurs secteurs industriels, l'étude de données spatio-temporelles pour des problèmes complexes est de plus en plus courante. Le développement de méthodes d'analyse pour des données de contrôle environnemental va donc permettre de faire avancer la science pour l'étude de tout type d'activités générant des données spatio-temporelles.

1.2. Question de recherche

La question de recherche qui découle de la problématique présentée est :

« Comment tirer parti de l'analyse multivariée et de la visualisation de données spatio-temporelles afin d'approfondir la compréhension du contrôle de la qualité environnementale des espaces de production pharmaceutique? »

1.3. Objectifs du projet

Objectif 1 : Analyser à l'aide d'outils mathématiques des données comportant des variations spatio-temporelles

1.1 Identifier et récolter les données historiques de différents types pouvant influencer le contrôle environnemental en milieu industriel.

1.2 Caractériser les différents types de données. Ceci débute par la détermination des données atypiques (incluant les données aberrantes ainsi que les données issues de conditions expérimentales moins rarement observées) et leur traitement (i.e. inclusion, exclusion, correction). On procède ensuite à la classification des données selon leurs types (quantitatif vs qualitatif) afin de nous enligner vers une stratégie d'organisation des données. Enfin, déterminer quelles variables présentent suffisamment de variabilité (aléatoire ou structurée) et qui pourront être utiles à la prédiction de déviations. Ceci revient à vérifier la variation en fonction du temps et de l'espace des données historiques.

1.3 Organiser les données spatio-temporelles de façon à appliquer des analyses pertinentes et identifier les paramètres importants du processus.

Dans le cas des données du processus environnemental, cet objectif se traduit premièrement par l'identification de l'information utile. La Figure 1.1 ci-dessous montre une représentation des données spatio-temporelles issues des analyses environnementales récoltées à différents endroits de l'usine (points rouges et verts) à plusieurs instances temporelles T.

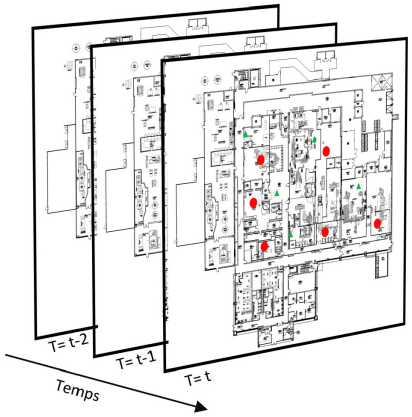

Figure 1.1 : Schématisation des données spatio-temporelles

Ensuite, il est nécessaire d'identifier les attributs de qualité mesurables pour la contamination bactérienne dans un milieu industriel (fréquence des tests positifs, magnitude des tests, seuils d'alerte). Finalement, pour procéder à une organisation représentative des données spatio-temporelles, il faudra aussi définir la résolution spatiale et temporelle d'intérêt. L'objectif spécifique aux données du projet est donc d'utiliser ces données spatio-temporelles afin d'analyser plus en profondeur le processus de contrôle environnemental.

<u>Objectif 2 : Faciliter l'interprétation des résultats des analyses spatio-temporelles à l'aide de modèles et d'outils adaptés</u>

2.1 Utiliser des analyses multivariées permettant d'extraire des tendances ayant un sens physique.

2.2 Identifier des outils de visualisation afin de facilement interpréter les dynamiques spatio-temporelles du processus à l'étude.

2.3 Présenter les résultats de façon simple et concise dans le but d'apporter des avenues d'investigations pour améliorer le contrôle environnemental.

1.4. Contributions originales

Ce projet de recherche contribue à la science en présentant de nouvelles approches pour l'analyse et la compréhension de jeux de données spatio-temporelles. Les contributions principales sont résumées ci-dessous :

- Développement d'un nouvel indice de similarité qui utilise une combinaison de la corrélation de Pearson et le « dynamic time warping » pour mieux traduire les dynamiques de contamination dans les données de contrôle environnemental.

- Développement d'une nouvelle approche multivariée pour l'étude de données spatio-temporelle fortement bruitée à l'aide de l'algorithme MCR-LLM (Multivariate Curve Resolution Log Likelihood Maximization).

- Outils de visualisation pour faciliter l'étude et l'interprétation des analyses sur les données de contrôle environnemental pour le partenaire industriel.

1.5. Plan du document

Le chapitre 2 résume l'état de l'art pour l'étude des données spatio-temporelles et plus spécifiquement les données de contrôle environnemental en milieu pharmaceutique.

Le chapitre 3 présente l'analyse par corrélation DTW et les outils de visualisation développés pour faciliter l'interprétation des résultats.

Le chapitre 4 présente un article soumis au journal Industrial & Engineering Chemistry Research pour l'identification de dynamiques et patrons à partir de données spatio-temporelles bruitées avec la résolution multivariée de courbe

Le chapitre 5 conclut sur le projet de recherche en présentant les bénéfices pour le partenaire industriel ainsi que les perspectives de travaux futures.

2. SYNTHÈSE DE L'ÉTAT DE L'ART

2.1. Contrôle environnemental en milieu industriel

L'enjeu des données présentant des variations spatio-temporelles concerne plusieurs milieux de production industriel. Il est notamment présent dans la prévention et le contrôle de la contamination, un enjeu majeur dans les secteurs alimentaires et pharmaceutiques. Le programme de contrôle environnemental constitue toutes les actions et les ressources que l'entreprise met en place afin de prévenir la contamination des produits. Dans le cadre du projet, l'analyse des données en lien avec le programme environnemental était à l'étude.

La majorité des tests microbiologiques des programmes de contrôle environnemental sont faits à l'aide de plaques de cultures et sont reportés en termes d'unité formant une colonie (CFU). Cette mesure est un indicateur du niveau de contamination qui se base sur le nombre de colonies, après incubation, à la suite d'un contact avec une surface testée.

La première dimension des analyses de contrôle environnementale est l'échantillonnage directement sur l'équipement du personnel (gants et habits de protection). Comme mentionné dans plusieurs études de la littérature [1][2], la majorité des contaminants retrouvés dans les milieux industriels sont généralement introduits par le personnel lors de l'entrée en usine et des activités de production. Ainsi, un équipement de protection individuel complet est utilisé dans les zones les plus à risques. La Figure 2.1 illustre ces habits dans le cas d'une usine pharmaceutique. Bien que ceux-ci diminuent significativement les risques de contamination, des défauts avec l'équipement ou des mauvaises pratiques individuelles peuvent mener à la contamination des zones avoisinantes et des produits fabriqués.

Ainsi, il est primordial de s'intéresser aux données des tests microbiologiques qui sont faits sur le personnel afin d'avoir un portrait représentatif de l'état des zones contrôlées. Cette information peut aussi être un bon indicateur des tendances à venir pour les autres surfaces des zones de production.

Figure 2.1 : Habit de protection du personnel dans un environnement pharmaceutique contrôlé [3]

Une seconde dimension très importante pour le contrôle environnemental est l'échantillonnage des surfaces critiques dans les zones de production. Par exemple, ce type d'analyse peut se faire sur le sol, les murs ou les surfaces de certains équipements. Ce type d'analyse se fait aussi avec des plaques de cultures et utilise l'unité CFU pour quantifier les résultats.

Finalement, l'échantillonnage de particules dans l'air a aussi une place importante dans les programmes de contrôle environnemental. Le but de ce type d'échantillonnage est à la fois de détecter les grosses particules inertes qui pourraient contaminer le produit, tout comme les particules viables (spores, bactéries) qui posent un risque majeur s'ils rentrent directement en contact avec celui-ci. Des équipements d'aspiration d'air sont utilisés pour la détection active de particules inertes alors que des plaques de culture exposée à l'air ambiant sont utilisées pour la détection passive.

Ces trois dimensions du contrôle environnemental constituent une partie importante des efforts que les usines mettent en place afin de prévenir la contamination des produits. À cela s'ajoute aussi la classification des différentes zones de productions selon 4 grades pharmaceutiques nécessitant des pratiques spécifiques [4]. En effet, les instances gouvernementales tel que la FDA exigent une classification des espaces de production en fonction du risque associé à une perte de contrôle environnemental. Une pièce dans laquelle un produit est exposé à l'air ambiant va généralement est classé comme à haut risque impliquant ainsi une fréquence d'échantillonnage plus élevée, une plus grande diversité de test et des nettoyages plus fréquents par exemple. Une description détaillée

des différents types d'analyse ainsi que des exigences reliées au différent pharmaceutiques est présentée en annexe.

Un des principaux défis avec l'analyse des données environnementales en milieu industriel est de déterminer comment interpréter correctement les résultats des tests microbiologiques. La rareté et la nature sporadique des événements de contamination constituent un défi majeur pour l'analyse des données. En effet, avec l'amélioration du contrôle bactérien des zones de productions industrielles, la majorité des analyses environnementales donnent des résultats nuls ou très faibles (en CFU). Avec une quantité limitée de résultats élevés, les patrons de contamination réels deviennent difficiles à distinguer du bruit provenant des conditions d'opération normales ou des contaminations croisées lors de la manipulation des plaques par exemple. De plus, même si les plaques de culture sont souvent représentatives des pires conditions dans une salle, un résultat CFU nul ne garantit pas l'absence de microorganisme puisque l'échantillonnage s'est fait uniquement sur une surface limitée. Les données de contrôle environnemental sont donc souvent caractérisées par un ratio signal sur bruit faible [5].

Les techniques d'analyse quantitative standards qui se basent uniquement sur les résultats quantitatifs de chaque test individuellement ne sont donc pas vraiment adaptées pour identifier les tendances et les situations problématiques. Plusieurs articles suggèrent donc d'utiliser la fréquence des résultats CFU positifs dans des zones contrôlées en parallèle avec l'amplitude de ces tests comme indicateurs [6], [7]. En effet, puisque le contrôle dans ce genre de salles est de plus en plus important, les résultats quantitatifs ne permettent plus d'aussi bien comprendre les dynamiques de contamination. En utilisant des techniques d'analyse qui se basent sur la fréquence et l'amplitude, on peut donc mieux interpréter les résultats et faire ressortir de l'information additionnelle.

L'importance de considérer la fréquence de détections des microorganismes est aussi confirmée dans la littérature qui évoque le phénomène de plateau [2] . En effet, une étude sur la contamination dans des salles contrôlées [8] a démontré que le même niveau de contamination est détecté sur des surfaces lisses après une semaine ou un an d'exposition dans ce genre d'environnement. Ceci est en effet dû à un manque de nutriments et aux dynamiques de survie des microorganismes sur ce genre de surfaces.

L'outil principal généralement utilisé aujourd'hui dans l'analyse des données de contrôle environnemental est la courbe de contrôle « control chart ». Cet outil permet de tracer les

résultats d'un test ou d'un groupe de tests en fonction du temps. L'analyse des courbes de contrôle se fait ensuite en traçant la moyenne ainsi que les limites supérieures basées sur les données historiques. En comparant les données récentes avec ces limites, on peut donc identifier des tendances ou des résultats problématiques qui suggèrent une déviation au contrôle environnemental. Quelques exemples d'utilisation de courbes de contrôle pour l'analyse environnementale en zone industrielle sont présents dans la littérature [6], [9]. L'article de Bar et al. [7] présente des méthodes simples pour la construction des courbes de contrôle en utilisant des moyennes mobile et la fréquence d'occurrence de tests positifs par exemple.

Mis à part les courbes de contrôle, la littérature est assez peu développée en ce qui a trait aux alternatives d'analyses et aux tentatives de modélisation du processus de déviation. Cependant, afin de correctement analyser le problème de contamination en zones contrôlées, il apparaît nécessaire de considérer les dynamiques spatiales (l'influence entre les pièces de l'usine) et temporelles (influence des tests et autres paramètres d'une journée à l'autre) du processus. Ceci revient donc à un problème de compréhension et de modélisation de données spatio-temporelles dans un contexte industriel. Une approche adaptée est donc l'utilisation de techniques d'analyse mathématique plus poussées propres aux études spatio-temporelles.

2.2. Données spatio-temporelles

Les données générées par le programme de contrôle environnemental forment une banque de données qui relie les différentes variables à des valeurs quantitatives/qualitatives avec de l'information supplémentaire par rapport à l'aspect temporel et spatial de chaque instance. La présence de cette information additionnelle amène plusieurs nouvelles possibilités pour l'analyse mais aussi des défis qui doivent être considérés pour correctement analyser et interpréter les données spatio-temporelles.

2.2.1. Considérations et caractéristiques

Une des caractéristiques les plus importantes des données spatio-temporelles est la présence de dépendance entre les observations en raison des relations spatiales et temporelles des données. Cette différence majeure avec des données sans caractéristiques temporelles ou spatiales fait en sorte que l'utilisation de techniques

d'analyse et de fouille de données standards peut conduire à de mauvais résultats ou une mauvaise interprétation de ceux-ci. Les considérations majeures à prendre en compte ont été étudiées dans la littérature [10][11] et les plus importantes sont résumées ci-dessous :

- Autocorrélation : Les observations ne sont pas indépendantes les unes des autres et dépendent généralement des observations proches spatialement et temporellement.

- Stationnarité : Les séries temporelles présentent généralement des défis additionnels dans l'analyse de données puisque la stationnarité n'est pas assurée. Une série temporelle stationnaire est une série qui a une moyenne et une variance constante pour les périodes temporelles d'intérêts. Cependant, pour des données spatio-temporelles qui décrivent des processus complexes, il est possible d'observer des périodes avec des variations de la moyenne ou de la variance. Il est donc nécessaire d'étudier et d'analyser ces données avec des méthodes adaptées qui prennent en compte la possibilité de non-stationnarité.

- Évaluation des résultats : L'évaluation des analyses ainsi que des résultats en modélisation de processus doit se faire différemment. En effet, des petites différences spatiales ou temporelles peuvent avoir un impact important sur la comparaison de deux séries lorsque les valeurs sont uniquement évaluées points à points. Des valeurs différentes mais proches au sens spatio-temporel (en raison d'un décalage entre deux séries par exemple) sont utiles à l'analyse, mais seront généralement rejetées avec des évaluations de performances conventionnelles. Cette évaluation doit donc prendre en compte les dynamiques spatio-temporelles des données.

2.2.2. Organisation des données spatio-temporelles

Il existe plusieurs approches pour organiser et analyser des données spatio-temporelles. En fonction du problème, il est important de définir la nature des observations et des variables. L'article d'Atluri [10] décrit bien les différentes façons d'organiser et d'analyser les données d'un problème spatio-temporel en fonction du type de données. Les principaux types de données spatio-temporelles (événements ponctuels, trajectoires,

données rasters pour de l'information associée à une position géographique) et la façon dont celles-ci peuvent être organisées en observation/instances (points, carte spatiale, séries temporelles) pour les analyses sont décrits. Dans le cas de données de contrôle environnemental, des analyses sont réalisées dans plusieurs pièces des zones de productions avec une fréquence temporelle variable. Une façon d'aborder le problème est donc d'utiliser une représentation des données sous forme de séries temporelles où chaque observation (instance) représente un point dans le temps pour lequel une analyse a été réalisée dans une pièce ou un endroit précis de la zone de production.

2.2.3. Type d'analyse

Plusieurs études dans les littératures répertorient les analyses possibles sur des données temporelles ou spatio-temporelles [10],[11]. En raison de la nature de données de contrôle environnemental, les analyses ci-dessous ont été étudiées.

- Prétraitements des données :

Une partie importante du prétraitement est le filtrage, la réduction du bruit et l'élimination des résultats aberrants. Avec des séries temporelles, une technique couramment utilisée pour réduire le bruit est de prendre une moyenne roulante sur une période de temps pertinente au problème. Cette opération a un effet de lissage et permet de réduire le bruit pour les analyses.

La détection des données aberrantes se fait principalement à l'aide de propriétés statistiques des séries temporelles et à l'utilisation de seuils. La variance ou l'écart type des séries sont souvent utilisés pour identifier les valeurs ne faisant pas partie de la distribution normale des données. Lorsque ces valeurs sont identifiées, plusieurs techniques existent pour les remplacer. Entre autres, il est possible d'utiliser la moyenne de la série ou d'effectuer une interpolation avec les valeurs avoisinantes. Le choix de la méthode pour remplacer les données aberrantes doit cependant se baser sur une compréhension des données.

La normalisation est aussi un aspect important de l'analyse des données de façon générale. En effet, lorsqu'on a des données avec plusieurs types de variables qui peuvent avoir des unités ou des ordres de grandeur différents, il faut souvent passer par une étape

de normalisation afin de ne pas donner un poids plus important à une variable qui varie dans un ordre de grandeur supérieur aux autres par exemple. Cette étape importante doit cependant être réalisée avec prudence. En effet, comme démontré dans l'article de Vlachos et al. [12], la normalisation d'une série temporelle avec beaucoup de bruit peut conduire à de mauvais résultats et à une mauvaise interprétation des données. En effet, si le bruit n'est pas pris en compte lors de l'étape de normalisation, celui-ci peut grandement influencer le calcul de la moyenne, de la variance ou des extremums.

Une autre étape très importante est l'imputation des données manquantes. Le choix de la technique utilisée pour gérer les données manquantes doit se baser sur plusieurs critères propres aux données tels que la fréquence des données manquantes, leur distribution ou les raisons d'absence de données. La littérature présente deux grands groupes de méthodes couramment utilisées pour l'imputation des données ; les méthodes statistiques et les méthodes par apprentissage [13]. Les techniques statistiques telles que la moyenne ou la régression linéaire sont les plus couramment utilisées en raison de leur simplicité. Les techniques par apprentissage sont quant à elles généralement plus complexes et nécessitent plus données, mais peuvent permettent d'obtenir de meilleurs résultats. Une bonne compréhension des données à l'étude est ainsi nécessaire pour correctement identifier une méthode d'imputation pour les données manquantes.

Finalement, il est possible que la résolution temporelle varie entre les tests ou dans le temps. Il est donc souvent nécessaire d'interpoler ou de sous-échantillonner certaines séries de données afin d'obtenir un ensemble uniforme pour la modélisation et les analyses subséquentes.

- Recherche par contenu :

Ce type d'analyse est souvent la première étape lors de l'étude des données. Elle consiste à utiliser des contraintes, des seuils, des transformations simples afin de sélectionner un sous-ensemble d'intérêt. Concrètement, ce type de recherche permet de se familiariser avec les données, d'identifier certaines tendances visuelles et d'avoir une idée des dynamiques spatiales et temporelles qui caractérisent les données.

- Regroupement et étude des relations :

Une approche couramment utilisée pour l'analyse de données spatio-temporelles est le regroupement en fonction de critères de similarités. L'objectif est d'identifier des groupes de localisations spatiales avec des séries temporelles qui présentent des tendances ou des variations corrélées. Différentes mesures de similarités ont été étudiées dans la littérature [14] et peuvent être utilisées en fonction du type de données et du type de relations recherchées. Entre autres, les techniques présentées les plus courantes sont :

- Distance euclidienne :

Ce critère de similarité se base sur une représentation des séries temporelles avec un point dans un espace euclidien à *n* dimensions. Les *n* dimensions correspondent au nombre d'instances temporelles de la série. Afin d'évaluer la similarité entre les 2 séries, il suffit alors de calculer la norme L_p avec *p*=*n* entre les deux points associés aux deux séries temporelles. Ce critère de similarité qui est très simple possède cependant plusieurs limitations pour des séries temporelles avec des relations un peu plus complexes. En effet, dans le cas où deux séries sont proches avec des variations similaires mais que l'entendue de ces variations est différente les comparaisons point à point de la distance euclidienne vont généralement donner de mauvais résultats. Même si la normalisation peut aider dans ce genre de situations, d'autres limitations apparaissent lorsqu'on a des variations avec des durées différentes selon l'axe temporel par exemple.

- Corrélation :

Un second critère de similarité couramment utilisé pour plusieurs applications est le coefficient de corrélation de Pearson. L'équation de ce coefficient est donnée ci-dessous :

$$\rho_{XY} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{2.1}$$

Avec *E* représentant l'espérance mathématique, X et Y des variables quelconques et μ et σ la moyenne et l'écart des variables utilisées.

Ce critère de similarité possède cependant lui aussi plusieurs limitations dans le cas des séries temporelles. Premièrement, lorsqu'on veut appliquer ce critère pour ce genre de séries, il est nécessaire de vérifier le respect de certaines conditions afin de bien appliquer

la méthode. Il faut en effet s'assurer de la stationnarité des séries temporelles en retirant les tendances et les saisonnalités afin de ne pas tirer de mauvaises conclusions en se basant sur des corrélations n'étant pas représentatives du système.

L'article de Erdem et al [15] expose une seconde limite du coefficient de corrélation standard en lien avec les propriétés statistiques du calcul du coefficient. En effet, les valeurs aberrantes ou des changements rapides de variations dans les données peuvent grandement influencer les résultats. On peut observer ce genre de comportement dans des situations comme celle illustrée à la Figure 2.2.

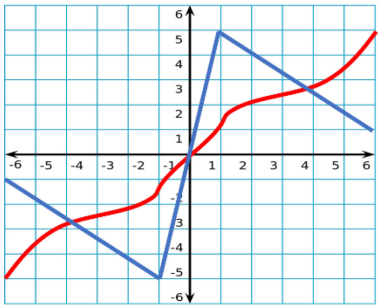

Figure 2.2 : Deux séries avec des variations locales opposées et un coefficient de corrélation de 0.5

Le calcul du coefficient de corrélation donne un résultat de 0.5 ce qui suggère une corrélation significativement positive. Cependant, le graphique permet de facilement remarquer les deux séries varient de façon opposée à l'exception du saut de valeur aux alentours de 0. Le résultat positif est dû au fait que la moyenne des deux séries est identique à 0. Ainsi, dans la partie négative de l'axe horizontal, les deux séries se retrouvent en dessous de leur moyenne ce qui va contribuer positivement au coefficient. Dans la deuxième partie, les deux séries vont alors être au-dessus de leur moyenne ce qui va encore contribuer positivement au coefficient. Ainsi, bien que ces deux séries varient de façon opposée à l'exception du saut de valeur, le coefficient de corrélation est significativement positif.

Afin de contrer ces limitations, une alternative au coefficient standard est présentée. Ce nouveau coefficient se base plutôt sur les variations entre les instances temporelles pour identifier des séries qui varient de façon similaire. Cette variante est donc mieux adaptée aux séries temporelles et l'équation pour le calculer est donnée plus bas.

$$\rho_{temporel} = \frac{E[(X_t - X_{t-1})(Y_t - Y_{t-1})]}{\alpha_X \alpha_Y} \quad (2.2)$$

et

$$\alpha_X{}^2 = E[(X_t - X_{t-1})^2] \quad (2.3)$$

Cependant, ce coefficient peut avoir beaucoup de difficulté à gérer les données avec un rapport signal/bruit faible. En effet, il est possible d'avoir des variations de $X_t - X_{t-1}$ qui ne traduisent pas la tendance réelle de la courbe si le bruit impacte de façon importante le calcul. Cet aspect est donc à prendre en compte. Une solution alternative au problème évoqué dans la Figure 2.2 est d'utiliser une moyenne roulante qui se met à jour dans le temps pour prendre en compte ces changements.

- DTW (Dynamic Time Warping) :

Un autre aspect à considérer lors de l'évaluation de la similarité entre 2 séries temporelles est la possibilité d'avoir des variations similaires mais désalignées. Cette situation peut se produire lorsqu'il y a des relations temporelles avec décalages dans les données. Dans ce cas de figure, les méthodes présentées précédemment auront beaucoup de difficulté à donner un indice de similarité représentatif. Une approche couramment utilisée est alors de définir une plage de délais temporels possible et de décaler les séries avec ces délais. Il suffit ensuite de choisir le délai qui permet d'obtenir la plus grande similarité entre les données pour trouver les délais optimaux qui reflètent les dynamiques temporelles et spatiales du processus. Ce type de relations sont en effet présentées dans les travaux de [16] qui a pour objectif la prédiction de précipitations futures en se basant sur les relations, avec délais temporels, entre les anomalies de température et de pression à la surface des océans. Cependant, il est aussi possible d'avoir des relations temporelles dynamiques avec des délais et distorsions variables. Ce type de problème est très fréquent dans le domaine de la reconnaissance vocale puisque le débit et les habitudes de paroles de chacun sont différents [17]. On fait donc face à des délais temporels optimaux intermittents qui sont difficiles à identifier en faisant une recherche gloutonne de tous les décalages possibles.

Afin de trouver les délais optimaux et les décalages pour plusieurs cas de figure, une approche par DTW (dynamic time warping) peut être utilisée. La Figure 2.3 démontre l'utilité du DTW en le comparant à la distance euclidienne afin d'évaluer la similarité entre

deux séries temporelles quelconques. La distance euclidienne est calculée à partir des relations points à points représentées par les lignes grises à la Figure 2.3(a). On peut rapidement se rendre compte que les deux séries ont des variations similaires mais que la distance euclidienne va donner un indice de similarité faible. L'évaluation de la similarité par DTW se base sur le calcul d'une matrice de ‘‘warping’’ illustrée à la Figure 2.3 (c) pour donner le chemin optimal afin d'optimiser les similarités points à points entre les deux séries. Ainsi, un nouvel indice va pouvoir être calculé à partir des relations optimales représentées par les lignes grises en (b). Afin de contraindre les décalages temporels possibles pour l'association, on peut utiliser un critère (représenté par les lignes noires) qui limite la déviation maximale point à point entre les deux séries temporelles. La distance calculée à partir du DTW va donc permettre d'obtenir des résultats beaucoup plus fidèles aux dynamiques réelles dans les données.

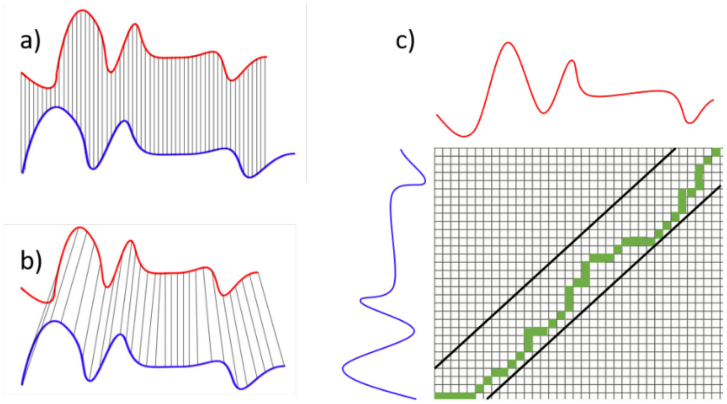

Figure 2.3 : Fonctionnement de l'algorithme DTW. En (a) les deux séries avant le réalignement, en (b) les deux séries à la suite du réalignement et en (c) la matrice de warping

Les méthodes simples d'évaluation de similarité telles que la distance euclidienne et le coefficient de corrélation ont donc plusieurs limitations avec des séries temporelles complexes pouvant avoir des décalages temporels ou certaines valeurs disparates. En effet, ce type d'analyse s'intéresse aux relations d'éléments précis un à un entre deux séries de données et ce sur toute la période de temps. L'algorithme DTW apporte une partie de solution à ce problème en considérant des plages de décalages possibles lors de l'évaluation de la similarité.

- Analyse de relations intermittentes

Il est aussi important de considérer la possibilité de relations intermittentes pouvant fortement impacter le calcul de la similarité avec les méthodes précédemment explorées. Une approche présentée par Atluri et al. [18] utilise cette notion pour identifier des périodes temporelles réduites de corrélation forte dans l'ensemble des données.

Pour identifier ces périodes, un coefficient de corrélation roulant est calculé sur des fenêtres temporelles mobiles afin d'obtenir plusieurs valeurs de corrélations différentes (associées à des sous-ensembles séquentiels) pour deux séries. Ensuite, un seuil de corrélation minimum est choisi afin de calculer le nombre de fenêtres qui présente un coefficient supérieur à celui-ci. Ce nombre de fenêtres permet alors de répertorier les paires de séries temporelles avec des relations intermittentes fortes difficilement repérables avec les méthodes conventionnelles. Il suffit ensuite d'identifier les fenêtres avec les coefficients élevés pour identifier les périodes en question.

Les approches plus complexes comme le DTW et l'analyse intermittente font généralement ressortir d'autres défis et limitations de l'identification des relations des données spatio-temporelles. En effet, en introduisant des calculs de relations intermittentes ainsi que des délais variables, on augmente considérablement le degré de liberté pour l'identification de ces relations. Ceci peut par conséquent mener à une mauvaise interprétation des résultats avec l'identification de relations éphémères et parfois aléatoires ne représentant pas le système dans son entier. Une approche permettant d'améliorer la robustesse des résultats et par conséquent limiter les fausses relations est l'analyse multivariée.

2.3. Méthodes multivariées

La motivation principale derrière l'utilisation d'analyses multivariées pour les données du projet est donc la quantité importante de données ainsi que les corrélations fortes attendues parmi certaines variables. De plus, une approche multivariée permet d'obtenir des résultats plus robustes en considérant toutes les variables simultanément plutôt que seulement des paires lors de l'analyse.

L'analyse MVDA (Multivariate Data Analysis) est une famille d'outils mathématiques en pleine expansion dans l'analyse de procédés industriels puisqu'elle permet d'interpréter une grande quantité de données en mettant en évidence l'interdépendance des différentes variables [19]. La décomposition par composantes principales (PCA) [20] est une méthode de réduction de dimension à la base de l'analyse multivariée. Elle permet de représenter les données originales avec un ensemble réduit de nouvelles variables orthogonales nommées composantes principales. Le principe derrière cette transformation est de trouver des hyperplans qui approximent les données avec un nombre réduit de dimensions qui maximisent la variance afin d'extraire des patrons de corrélation facilitant l'interprétation des données.

La transformation de la matrice de données initiales X (M $\times$ N) en une nouvelle matrice T (M $\times$ K) réduite se fait avec les composantes principales p compris dans la matrice P (N $\times$ K). On obtient donc la transformation suivante :

$$\mathrm{X} = \mathrm{TP}^{\mathrm{T}} + \mathrm{E} \tag{2.4}$$

avec E (M $\times$ N) la matrice résiduelle représentant l'erreur d'approximation des données originales par le modèle linéaire réduit. La matrice T contient les valeurs des différentes observations de la matrice initiale sur l'hyperplan construit à partir des composantes principales de P. Concrètement, les composantes principales p définissent le nouvel hyperplan et donnent de l'information sur les variables initiales qui contribuent le plus à l'orientation de ceux-ci, c'est-à-dire les variables les plus importantes pour la maximisation de la variance.

Cependant, une des limitations importantes de PCA est qu'il a pour seul objectif de maximiser la variance lors du calcul des composantes principales. Ainsi, les composants obtenus n'ont pas nécessairement une interprétation physique pertinente à l'analyse des données.

L'analyse MCR (Multivariate Curve Resolution) [21] est une alternative à la décomposition PCA qui apporte une solution pour l'obtention de composants ayant une interprétation significative. Cette méthode permet tout d'abord d'introduire des contraintes spécifiques aux données analysées lors du calcul des composants [22]. Ceci permet d'obtenir des résultats plus facilement interprétables ayant un sens physique.

L'algorithme MCR standard permet d'effectuer une décomposition par régression ALS (Alternating Least Square). MCR-ALS réduit et décompose les données initiales avec un modèle linéaire qui estime une matrice de données D (M $\times$ N) avec une matrice de contribution réduite C (M $\times$ K) et une matrice de poids S (N $\times$ K). Le modèle linéaire peut être écrit sous la forme :

$$D = C S^T + E \tag{2.5}$$

Le modèle est donc très similaire à PCA mais diffère en introduisant des contraintes dans les étapes de régressions. Ces contraintes, comme la non-négativité, la fermeture ou la symétrie permettent d'influencer le calcul des matrices C et S [23] afin d'obtenir des résultats plus facilement interprétables et d'améliorer la robustesse du modèle.

L'autre aspect qui contribue à l'obtention de résultats plus facilement interprétables est la nature non imbriquée de l'algorithme. À la différence de PCA, lorsqu'on calcule les nouvelles composantes principales, on va aussi mettre à jour celles précédemment calculées afin de mieux répartir la variance. Ceci permet d'obtenir des résultats faisant plus de sens mais rend aussi l'algorithme plus sensible au choix initial du nombre de composants pour la décomposition. Pour ce faire, des techniques comme le calcul du rang effectif d'une matrice permettent d'estimer le nombre de composants à utiliser pour la décomposition [24].

L'algorithme MCR-ALS a entre autre été utilisé avec succès pour l'identification de patrons de contamination avec des données spatio-temporelles pour des analyses de qualité de l'eau [25][26][27]. Des résultats intéressants ont aussi été obtenus pour l'analyse de données de pollution de l'air dans des zones urbaines [28].

Cependant, il a été démontré que MCR-ALS a beaucoup de difficulté à gérer les données bruitées. En effet, les étapes de régression de l'algorithme ALS ne sont pas adaptées à des jeux de données avec un ratio signal sur bruit faible et introduisent souvent des erreurs pouvant nuire à la convergence de l'algorithme. Des études sur des données spectrales fortement bruitées ont mis en évidence les limitations de l'algorithme MCR-ALS [29][30][31]. Il a par exemple été démontré que l'algorithme n'était pas en mesure de bien séparer les différents patrons à partir de d'images spectrales bruitées [29], compliquant ainsi l'interprétation des résultats

En raison du bruit important attendu dans les données de contrôle environnemental, une approche qui considère cet aspect est donc nécessaire. MCR-LLM (Log Likelihood Maximization) a récemment été présentée comme une alternative à MCR-ALS pour l'analyse d'images spectrales bruitées [29]. Cette variante passe par une étape de maximisation d'une distribution de Poisson plutôt qu'une régression linéaire afin de calculer la matrice réduite C. Cette approche a ainsi permis d'obtenir des résultats beaucoup plus représentatifs et facilement interprétables pour les données étudiées [29][31].

La modélisation à partir de méthodes multivariées permet d'apporter une solution au problème des données externes multiples avec la possibilité de corrélations fortes. De plus, elle permet d'obtenir des résultats plus robustes en se basant sur des groupes de variables afin de faire ressortir des tendances et des patrons dans les données. Cependant, ce type de modélisation se base principalement sur des modèles linéaires. En raison de la nature des données de contrôle environnemental, il est possible que certaines relations ou dynamiques soient difficilement capturables avec ce type de modèle. Une approche de modélisation par apprentissage machine a donc été envisagée.

2.4. Apprentissage Machine : Réseaux Neuronaux

A la base de l'apprentissage machine, les réseaux neuronaux sont utilisés depuis plus de 20 ans pour l'analyse de données complexes avec des relations non linéaires. Les réseaux de neurones parviennent à modéliser des relations non linéaires complexes à l'aide de fonctions d'activation de type sigmoïde ou tanh (tan hyperbolique) qui relient des entrées et une sortie selon une fonction non linéaire. L'apprentissage du réseau se fait en posant des poids préliminaires associées aux différentes entrées et en choisissant une fonction d'activation qui va renvoyer une sortie précise en fonction des entrées et des poids qui leur sont associés. Si les fonctions d'activation n'étaient que linéaires, les réseaux neuronaux se comporteraient comme des régressions linéaires multiples. Les différentes fonctions d'activation non linéaires ont donc comme avantage de permettent la modélisation de relations non linéaires entre les entrées et les variables de sortie.

Les réseaux neuronaux permettent ainsi de reconnaître des motifs et des tendances dans des données difficilement modélisables avec des méthodes linéaires. Un exemple

d'application sur des données de consommation d'énergie datant de plusieurs années est présenté dans l'article de Hippert [32]. L'auteur démontre les limitations des techniques de régression classiques et l'utilité de la modélisation par réseau de neurones lorsqu'on est en présence de relations non linéaires avec des variables environnementales.

En raison du grand intérêt dans les dernières années pour les algorithmes d'apprentissage machines, le développement de nouveaux algorithmes de réseaux neuronaux s'est fait de façon très importante. Une variante adaptée à la prédiction de séries temporelles est le réseau de neurones récurrents RNN (Recurring Neural Network) basé sur les travaux de Rumelhart [33]. Les modèles RNN sont en effet une variante aux réseaux de neurones standards qui incluent une boucle de rétroaction pour modéliser l'information des délais temporels passés. Pour ce faire, les sorties des neurones dans les couches cachées peuvent être gardées en mémoire et utilisées comme une entrée additionnelle dans le calcul du neurone suivant. Une extension aux réseaux neuronaux récurrents qui est devenue la norme pour ce type d'applications est le réseau neuronal LSTM (Long Short Term Memory). Cette variante introduite par Hochreiter and Schmidhuber [34] amène une solution aux problèmes des RNN lorsqu'on a des dépendances temporelles avec des délais plus longs [35]. Les neurones LSTM possèdent une complexité supplémentaire qui leur permet de prendre des décisions par rapport à l'information passée à considérer et celle à oublier. L'approche LSTM permet donc d'avoir une architecture plus stable pour les relations temporelles plus complexes et à plus long terme.

Une application récente des réseaux LSTM décrite dans la littérature est la prédiction de la qualité de l'air des jours à venir en se basant sur les données passées ainsi que des variables externes météorologiques [36]. L'étude présente une méthodologie qui combine les méthodes multivariées et la modélisation par apprentissage machine LSTM. Dans un premier temps, la décomposition PCA est utilisée afin d'identifier les variables les plus importantes et donc réduire le nombre de variables tout en conservant un maximum de variance. Ce nouvel ensemble de données est ensuite fourni à l'algorithme LSTM pour la prédiction des jours à venir. Bien que ce problème ne prenne pas en compte de dynamique spatiale, une méthodologie similaire pourrait être efficace avec les données de contrôle environnemental.

Une étude basée sur un modèle LSTM pour la détection d'erreurs dans un processus industriel est présentée dans l'article de Filonov [37]. Cette application se rapproche de la problématique du projet de maîtrise puisque l'auteur utilise le modèle pour la prédiction de déviations futures d'un procédé industriel de production de diesel. L'étude s'intéresse plus précisément à la modélisation du procédé avec des séries temporelles multivariées. Cependant, en raison de la présence très faible d'erreurs dans ce type de données, des séries artificielles ont été générées pour entraîner le modèle. En effet, une des limitations majeures de l'approche par apprentissage machine est la nécessité d'avoir beaucoup de données d'entrainement afin d'avoir un modèle final robuste sans surapprentissage.

Cette problématique est encore plus réelle pour les données de contrôle environnemental puisque les déviations et les périodes de contamination sont très rare dans les zones pharmaceutiques contrôlées. De plus, comme la fréquence des tests microbiologiques est rarement supérieure à un test par jour dans chaque pièce, le nombre total de données pour chaque série temporelle est assez limité. Ces deux caractéristiques importantes font en sorte que les techniques de modélisation par apprentissage machine ne sont pas vraiment adaptées. Pour ces raisons, l'analyse par apprentissage machine n'a pas été explorée plus profondément.

2.5. Conclusion sur l'état de l'art

La littérature offre donc plusieurs approches et solutions pour l'analyse des données de contrôle environnemental en milieu industriel. Il est tout d'abord important de prendre en compte les caractéristiques spécifiques aux données telles que le rapport signal sur bruit faible et l'incertitude en lien avec les tests microbiologiques.

De plus, la composante spatio-temporelle introduit des considérations additionnelles telles que la stationnarité et l'autocorrélation.

Des indices de similarité tels que la corrélation et le « dynamic time warping » (DTW) peuvent permettre d'analyser des données spatio-temporelles en faisant ressortir les patrons de corrélation et les groupes avec des variations similaires.

L’analyse multivariée avec PCA ou MCR peut elle aussi extraire des patrons de façon robuste et mettre en évidence des dynamiques spatiales et temporelles pour faciliter l’interprétation des données.

Les approches par apprentissage machine ne sont pas bien adaptées à la nature des données en raison de la quantité importante de données requises.

Finalement, comme mentionnées dans l’article de Keogh [38], les approches présentées dans la littérature ne doivent être qu’un point de départ, car les résultats obtenus sur des données synthétiques ou d’autres ensembles de données ne sont pas toujours représentatifs des résultats pour un nouveau jeu de données. Une analyse spécifique aux données de contrôle environnemental est donc l’approche à favoriser pour faire avancer la science dans le domaine pharmaceutique.

3. ANALYSE DES DONNÉES DE CONTRÔLE ENVIRONEMNTAL

3.1. Données utilisées

Les données de contrôle environnemental sont composées des résultats des analyses microbiologiques qui sont faites dans les zones de production d'une usine pharmaceutique sur une période de 3 ans. Plusieurs types de tests sont effectués (analyse au sol, sur le personnel, équipements, particules dans l'air) générant ainsi des données avec des unités et des ordres de grandeur différents. De plus, à chaque test sont associées une date et une position dans l'usine pour former un ensemble de données spatio-temporelles.

Dans le cadre du projet, les données entre 2015-07-01 et 2018-07-01 ont été conservées pour un seul bâtiment du site de production. Les principales analyses utilisées dans le cadre du programme de contrôle environnemental sont décrites plus en détail dans le Tableau 3.1 ci-dessous.

Tableau 3.1 : Description détaillée des principales analyses effectuées dans le cadre du programme de contrôle environnemental

Type d'échantillon	Méthode	Équipement	Unité
Microorganismes dans l'air	Passive (SETTLE)	Plaque de culture	CFU/plaque
	Active (VAIR)	SAS air sampler	CFU/FT^3
Microorganismes sur les surfaces	Plaque de contact	Plaque RODAC	CFU/25cm^2
Microorganismes sur le personnel	Plaque de contact	Plaque RODAC	CFU/arm
Particules dans l'air	Compteur de particules	Climet	Compte

L'indicateur de contamination qui a été choisi pour les analyses sur plaques de culture est le nombre de tests positifs par jour. Une plaque avec au moins une colonie (en termes de CFU) est considérée comme un test avec un résultat positif alors qu'une plaque sans croissance est considérée comme négative. En interprétant les résultats de façon binaire, on obtient un indicateur qui reflète la fréquence d'occurrence d'échantillons positifs. Ce choix s'est fait en se basant sur la revue de littérature qui a démontré que les résultats

quantitatifs issus du décompte de colonies bactériennes sur des plaques de cultures se retrouvent généralement dans la zone de bruit avec peu de colonies.

En effet, puisque la majorité des résultats quantitatifs (en termes de CFU) sont nuls ou très faibles, la fréquence des tests positifs est un indicateur plus représentatif des dynamiques de contamination. Le terme utilisé afin de référer cet indicateur est CRR (Contamination Recovery Rate).

Pour les tests du nombre de particules dans l'air, l'indicateur choisi a été la valeur moyenne quantitative normalisée par jour. En effet, comme ce type de test détecte tous types de particules d'une grosseur supérieure à 0.5 micron, les résultats quantitatifs sont souvent supérieurs à zéro. La valeur quantitative permet donc de plus fidèlement traduire les dynamiques de contamination en lien avec les particules dans l'air dans la zone de production.

La Figure 3.1 ci-dessous représente un plan schématique de la zone aseptique de production du partenaire industriel. Par souci de confidentialité, les noms des pièces ont été changés. Les différentes flèches représentent les portes pour entrer dans la zone aseptique (en vert), pour se déplacer entre les pièces (en bleu) et pour quitter l'espace contrôlé (en rouge). L'icône humain identifiée par la variable PM fait référence aux échantillons provenant des analyses sur le personnel.

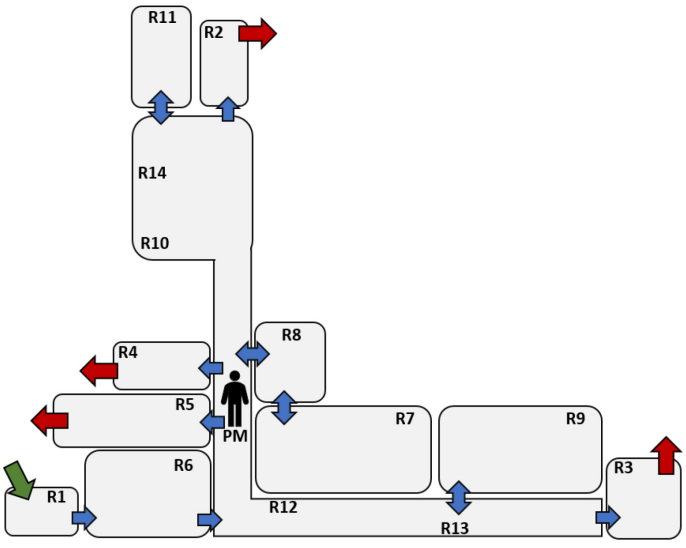

Figure 3.1 : Carte de la zone aseptique de production

Les zones de production pharmaceutiques sont généralement classées selon une échelle de grade entre D et A en fonction du risque de contamination du produit. Par exemple, une pièce de grade A nécessite des mesures beaucoup plus strictes et est échantillonnée de façon plus fréquente afin d'assurer le contrôle environnemental. Ceci se traduit donc par une fréquence d'échantillonnage étant différente d'une pièce à l'autre et pouvant varier d'une période de l'année à une autre.

Afin d'illustrer ce point, le nombre total de tests effectués dans les différentes pièces du jeu de données environnementales a été calculé. Les tests effectués sur le personnel ont été identifiés avec le sigle 'PM'. La Figure 3.2 illustre la disparité du nombre de test en fonction des pièces en montrant le nombre total de tests pour les 30 pièces les plus testées. Certaines de ces pièces ne se retrouvent pas dans la zone aseptique de l'espace de production ce qui explique leur absence sur le schéma de la Figure 3.1. La manipulation des données s'est effectuée avec le langage de programmation Python à l'aide de plusieurs librairies pour l'analyse de données tel que Pandas et Numpy.

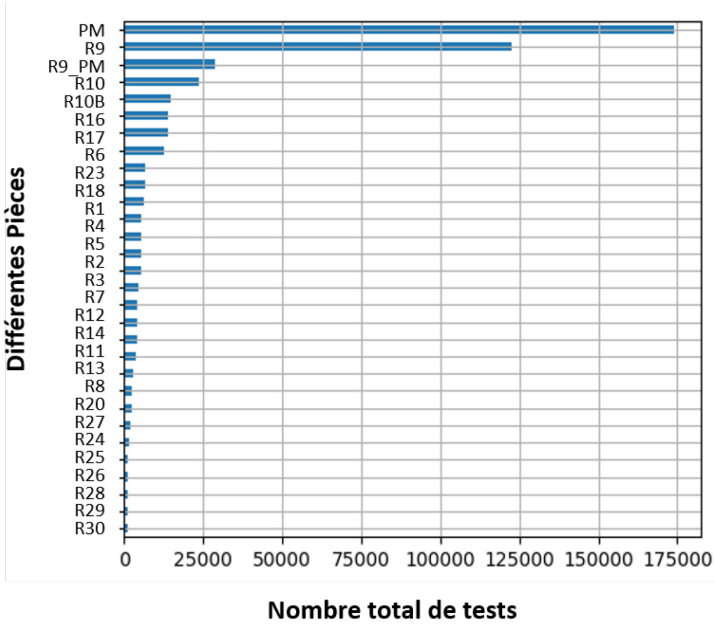

Figure 3.2 : Nombre total de tests effectués dans les différentes pièces du jeu de données

Comme mentionner précédemment dans Tableau 3.1, plusieurs analyses différentes sont effectuées dans les pièces de l'usine afin des vérifier les différentes composantes du contrôle environnemental. L'allure générale des résultats des tests de contrôle environnemental d'une pièce typique (R9) est illustrée à la figure Figure 3.3. En (a) est montré l'indicateur CRR pour les différents types de tests avec le CFU comme unité. Les variables associées au nombre normalisé (écart-type) de particules plus grandes que 0.5 et 5 microns sont montrés en (b).

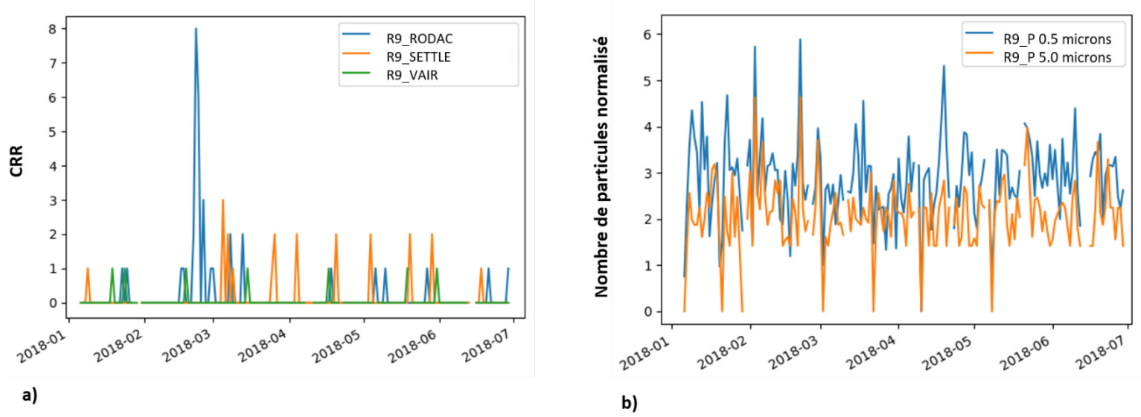

Figure 3.3 : Exemple de série temporelle des différents types de tests dans une pièce de la zone de production. En (a) pour les analyses CFU et en (b) pour le nombre de particules dans l'air

L'exemple des résultats des analyses de la pièce R9 permet d'illustrer la fréquence très faible de résultats CRR positifs pour une durée d'un an. Comme évoqué dans la littérature, ceci est dû au contrôle environnemental très strict dans les zones de production. Pour le nombre de particules, on peut se rendre compte que les indicateurs quantitatifs sont majoritairement supérieurs à zéro et permettent donc de mieux traduire les dynamiques de contamination qu'un indicateur binaire.

Les données historiques pour le bâtiment étudié sont composées d'un total de 48 pièces différentes avec 5 types d'analyses distinctes. Ces analyses sont : les analyses sur plaque RODAC pour les surfaces et le personnel, les analyses passives de microorganismes dans l'air avec plaque de culture, les analyses actives de microorganismes dans l'air avec un échantillonneur d'air et les analyses de particules inertes pour celles supérieures à 0.5 et celles supérieures 5 microns. Les données spatio-temporelles forment donc une matrice 3D de dimension $(1016 \times 48 \times 5)$ avec 1016 journées testées. Cependant, comme la fréquence d'échantillonnage et le type d'analyses effectuées varient d'une pièce à

l’autre, la matrice 3D comporte beaucoup de données manquantes.

Afin d’avoir une idée plus représentative du pourcentage de données manquantes, la matrice de données 3D a été réorganisée en supprimant la dimension spatiale et en renommant les variables avec l’ajout de la pièce associée. La nouvelle matrice 2D de dimension (1016 $\times$ 182) comportait alors 55.65% de données manquantes. Une représentation visuelle de cette matrice avec les données manquantes en blanc est illustrée à la figure Figure 3.4.

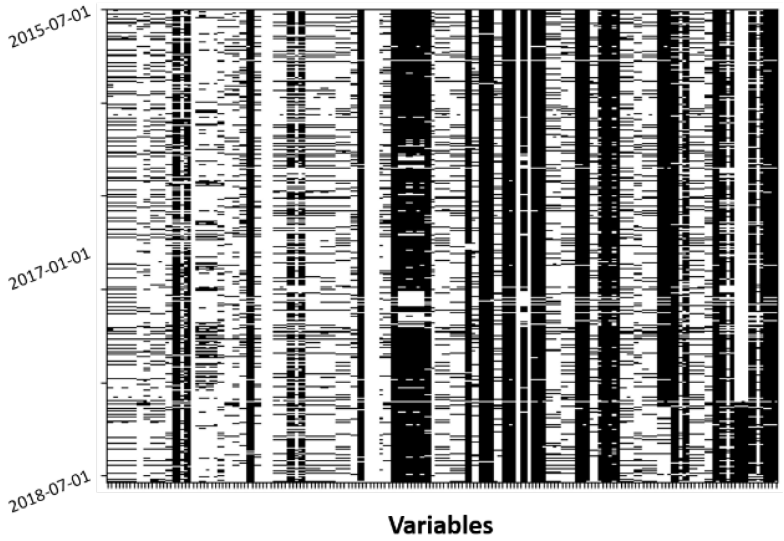

Figure 3.4 : Représentation visuelle des données manquantes avec chaque colonne étant associée à une variable et chaque ligne une journée. Les données manquantes sont représentées en blanc.

Une analyse visuelle de cette matrice permet de constater que la distribution de données manquantes est pratiquement binaire. En effet, une variable possède soit un pourcentage très élevé de données manquantes (colonnes avec une proportion de blanc beaucoup plus importante) ou très peu données manquantes (majoritairement noire). Les étapes de prétraitement vont donc pouvoir facilement retirer les variables avec un pourcentage très élevé de données manquantes et uniquement conserver les variables d’intérêts.

3.2. Prétraitements

Plusieurs étapes de prétraitement ont été utilisées afin de nettoyer, filtrer et organiser les données de contrôle environnemental pour les analyses. Ces étapes permettent entre autres d'éliminer les erreurs d'entrées, les résultats aberrants, les périodes non testées et les analyses supplémentaires en réponse aux alertes afin de s'assurer que les données analysées soient représentatives des dynamiques de contaminations.

La première étape a été de nettoyer les données brutes en renommant certaines variables, en harmonisant les unités et uniformisant la fréquence d'échantillonnage afin d'avoir une seule valeur journalière pour chaque variable. Le sous-échantillonnage (lorsque nécessaire) s'est fait avec la somme des analyses de même nature pour une journée.

Ensuite, les pièces et les analyses avec une proportion de données manquantes supérieure à 50 % ont été retirées. Les jours avec plus de 50% d'analyses manquantes ont aussi été retirés. Puisque les données manquantes ont par la suite été remplacées, ce choix a été fait afin d'éviter d'inclure des variables avec une proportion plus grande de données synthétiques par rapport aux données réelles. Ceci pourrait en effet biaiser les analyses en fonction de l'algorithme utilisé pour le remplissage des données.

En raison de la nature sporadique des événements de contamination ainsi que la faible proportion de résultats positifs en termes de CRR, les différentes analyses avec comme unité le CFU et effectuées dans une même pièce ont été additionnées pour chaque pièce dans la zone de production. Ce nouvel indicateur représente ainsi le niveau de contamination global journalier pour chaque pièce. Les variables quantitatives associées au nombre de particules dans chaque pièce ont été conservées séparément en raison des unités et des plages de valeurs différentes.

À cette étape du prétraitement, trois types de variables différentes sont présentes. Les variables associées au niveau de contamination globale d'une pièce sont étiquetées par le numéro de la pièce associée (R9 par exemple). Les variables associées au nombre de particules dans l'air pour les différentes pièces sont aussi étiquetées avec le numéro de la pièce avec l'ajout d'un suffixe permettant de les différentier des variables CRR (R9_P par exemple). Finalement, la variable associée à la contamination du personnel est

étiquetée avec l'abréviation 'PM'.

Une somme roulante sur une période de 7 jours a ensuite été appliquée aux données afin de mieux capturer les tendances de contamination et donner moins d'importance aux résultats positifs isolés temporellement. L'analyse des données de contrôle environnemental sur un horizon minimal d'une semaine permet de distinguer les périodes avec des tendances de contamination de celles sans contamination ou avec des détections sporadiques et isolées. Un horizon supérieur à une semaine entraîne une réduction significative de la résolution spatiale des analyses. La somme roulante sur 7 jours permet ainsi de mieux répondre aux objectifs du projet et d'obtenir un indicateur plus robuste qui prend en compte l'information temporelle adjacente. La Figure 3.5 illustre les différentes analyses séparément ainsi que l'indicateur global roulant pour la pièce R9 après le regroupement sous un seul indicateur global et l'opération de la somme roulante.

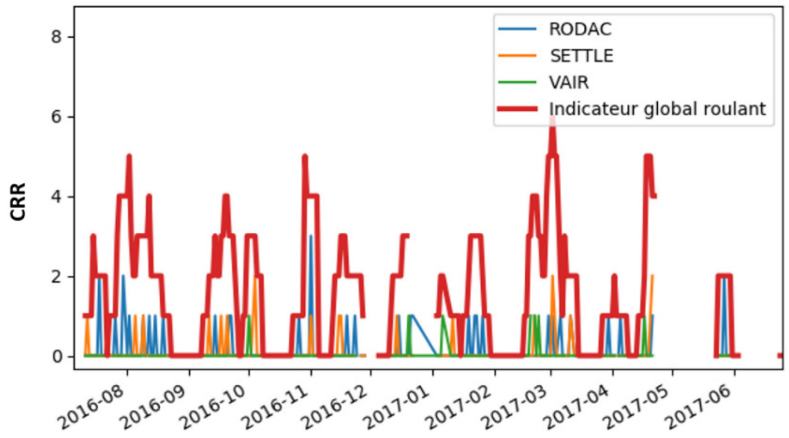

Figure 3.5 : Résultats des analyses microbiologiques séparées et indicateur global roulant pour la pièce R9

Les données aberrantes, en lien avec des erreurs d'entrées par exemple, ont ensuite été retirées à l'aide d'un seuil basé sur la variance. Les valeurs supérieures à trois fois la variance des séries temporelles ont été ramenées vers la moyenne.

En raison de la diversité du type de tests et des endroits testés dans les données de contrôle environnemental, les différents indicateurs de contamination ont été normalisés. Comme les différentes variables peuvent avoir des unités ou simplement des plages de valeurs différentes, la normalisation assure la cohérence des analyses.

Une pondération des indicateurs de contamination a aussi été ajoutée en fonction des pièces afin de faire le lien avec la fréquence d'échantillonnage. En effet, ceci a été fait pour éviter qu'un poids trop important soit accordé aux pièces avec très peu de tests et donc peu de résultats positifs après l'étape de normalisation. Les poids ont été calculés avec la racine cubique du nombre total de tests effectués dans chaque pièce. Ce facteur d'échelle est arbitraire et doit être adapté ou enlevé pour d'autres jeux de données. Les analyses préliminaires ont démontré que l'utilisation d'une telle pondération permet d'augmenter le poids des pièces avec une fréquence d'échantillonnage élevée et de diminuer celui des pièces peu testées. En raison du ratio signal sur bruit faible des données de contrôle environnemental, les résultats des pièces avec peu de tests ont été considérés comme moins fiables, justifiant ainsi l'utilisation du facteur de pondération.

Les données manquantes restantes ont ensuite été remplacées à l'aide d'un modèle PCA à 5 composants en se basant sur un rang effectif calculé de 5.8. Le calcul du rang effectif permet d'estimer le rang d'une matrice en considérant uniquement les valeurs propres significativement différentes de zéro. En choisissant un modèle à 5 composants, on peut donc avoir une bonne estimation des valeurs manquantes.

Finalement, la matrice de données 3D spatio-temporelles a été réorganisée de la même façon que lors du calcul des données manquantes. L'axe spatial a été supprimé et les indicateurs pour chaque pièce ont été renommés en leur ajoutant le nom de la pièce. La nouvelle matrice 2D après toutes les étapes de prétraitement était donc de dimension (988×24) avec 24 variables représentant le niveau de contamination et le nombre de particules dans les différentes pièces pour un total de 988 jours.

3.3. Corrélation par déformation temporelle dynamique

Pour évaluer les relations entre les différentes variables de l'ensemble de données, un indice de similarité qui utilise le Dynamic Time Warping (DTW) ainsi que la corrélation de Pearson a été développé. L'objectif derrière cet indicateur est de prendre en compte des délais temporels variés et changeants entre les variables qui pourraient considérablement nuire à un coefficient de corrélation standard.

La première étape avant d'évaluer les similarités est la vérification de la stationnarité des différentes séries temporelles dans les données. Une série temporelle stationnaire est une série qui a une moyenne et une variance constante pour les périodes temporelles d'intérêts. Cette caractéristique est importante afin d'éviter la détection de fausses corrélations en raison d'une tendance à la hausse pour 2 séries par exemple. Le test augmenté de Dickey Fuller est un test statistique avec hypothèse qui permet de vérifier si une série temporelle est stationnaire [39]. Pour ce faire, le test a comme hypothèse nulle que la série n'est pas stationnaire. Afin de réfuter cette hypothèse initiale et prouver la stationnarité d'une série, la *p*-value calculé doit être inférieur à un certain seuil significatif généralement posé à 0.05. Le Tableau 3.2 donne les 10 valeurs les plus élevées du test statistique de Dickey Fuller pour les séries temporelles des données de contrôle environnemental. Les variables avec une lettre P représentent le nombre de particules dans les pièces associées.

Tableau 3.2 : Valeurs du test statistique de stationnarité pour les différentes variables

Variable	*p*-value
PM	0.01684
R1	0.000296
R2	2.67E-05
R10_P	9.29E-06
R3	8.73E-06
R8	5.91E-06
R5_P	4.25E-06
R8_P	2.46E-06
R5_P	1.22E-06
R9	1.18E-06

Ce test permet d'affirmer que toutes les variables sont stationnaires avec une confiance supérieure à 95 %. Le tableau complet avec les valeurs du test de stationnarité pour toutes les variables se retrouve en annexe.

Dans le but de démontrer l'utilité ainsi que la méthodologie utilisée pour le calcul de la corrélation DTW, deux séries temporelles provenant des données de contrôle environnemental ont été utilisées. La première est le niveau de contamination CRR pondéré pour les tests effectués sur le personnel (PM) et la deuxième est le niveau de contamination CRR global et pondéré pour la pièce R9. La Figure 3.6 illustre les variations temporelles des deux séries sur une période d'environ 2 mois. Cette période a été choisie, car elle démontre bien la différence entre la corrélation standard et la corrélation DTW.

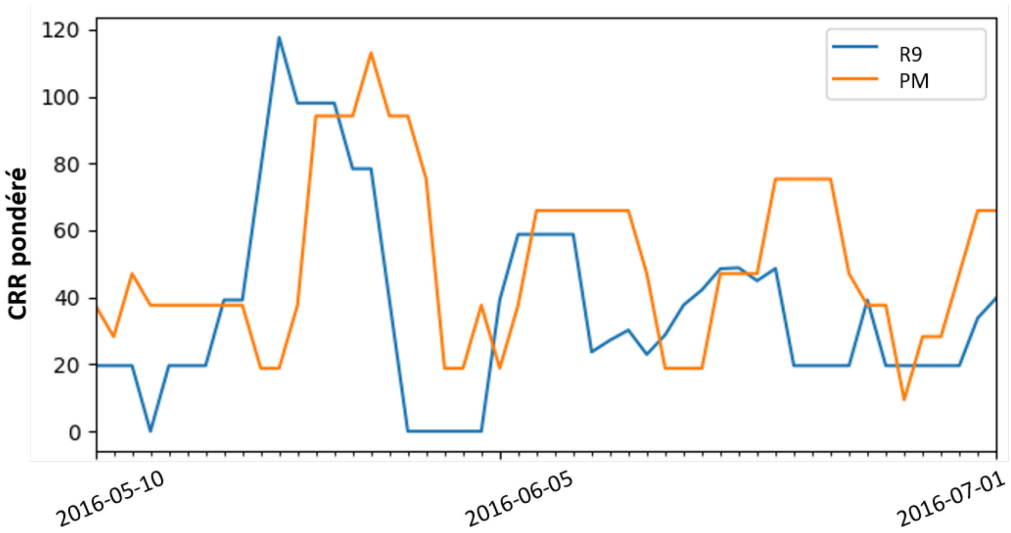

Figure 3.6 : Visualisation des indicateurs de contamination du personnel et de la pièce R9 pour une période de 2 mois

En observant les variations des deux séries, on se rend compte qu'il y a des tendances et des patrons de variation similaires entre celles-ci. En effet, l'allure générale des indicateurs de contamination semble indiquer qu'il y a une corrélation significative entre la pièce et la contamination du personnel. Cependant, comme les pics et les creux ne sont pas parfaitement alignés, le coefficient de corrélation de Pearson standard donne une valeur relativement faible de 0.23. Ce genre de décalage peut se retrouver dans les données de contrôle environnemental pour plusieurs raisons. En effet, il dans un premier temps possible que la contamination de l'un provoque celle de l'autre. Il est alors normal d'Avoir un petit décalage entre les pics des deux séries. Une autre raison possible

expliquant des décalages est l'incertitude en lien avec la détection des microorganismes. Comme une quantité limitée de surfaces sont testées chaque jour, il est possible qu'un contaminant soit uniquement détecté le lendemain par exemple. Un indice de similarité qui se base sur le coefficient de corrélation a ainsi beaucoup de difficulté à gérer les relations avec des délais temporels intermittents et variés. Ceci explique l'obtention d'une valeur qui reflète mal les dynamiques réelles dans les données.

Le calcul de la corrélation DTW se base dans un premier temps sur l'algorithme de DTW et l'obtention de la matrice de ''warping''. La matrice de warping illustrée à la Figure 3.7(a) donne le chemin optimal (en rouge) afin de maximiser les similarités point à point entre les deux séries. Ce chemin est obtenu en déterminant les décalages temporels optimaux, respectant la contrainte de décalage, pour minimiser les distances point à point entre les deux séries. La contrainte de décalage fait varier le degré de liberté de l'algorithme en permettant un réalignement plus ou moins important entre les deux séries. Cette contrainte est représentée par les cases colorées sur la diagonale de la matrice de warning. Dans le cadre des données de contrôle environnemental, un critère de décalage temporel maximal de 2 jours a été utilisé afin de conserver une interprétation physique cohérente des résultats. En choisissant la borne inférieure de 2 jours, des pics de contamination fortement décalés et sans lien réels ne seront pas alignés. Ceci permet donc de conserver uniquement les relations les plus robustes et ayant un sens physique. La Figure 3.7(b) montre les 2 indicateurs de contamination avec les nouvelles correspondances point à point (en gris) obtenus par l'algorithme de DTW.

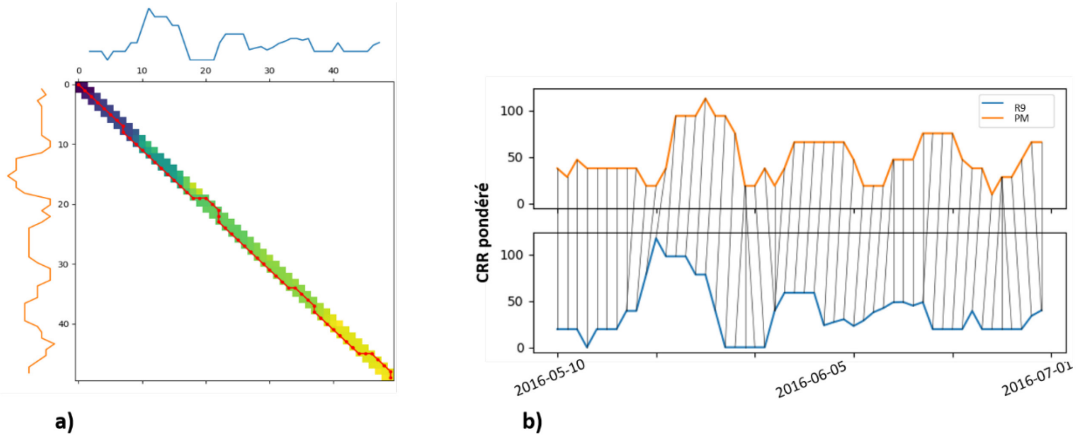

Figure 3.7 : Matrice de ''warping'' (a) et décalages temporels optimaux (b) pour les deux indicateurs de contamination

Cette information permet ensuite d'effectuer un réalignement des deux séries temporelles afin d'avoir une correspondance point à point plus représentative des dynamiques de contamination. La Figure 3.8 montre les deux mêmes séries temporelles à la suite de l'étape de réalignement. Sans changer l'allure globale des deux séries, les pics et les creux sont maintenant mieux alignés. Le coefficient de corrélation DTW est de 0.53. L'alignement entre les deux séries n'est pas parfait en raison de la contrainte maximale de warping qui limite les décalages à 2 jours. Cette contrainte est cependant nécessaire afin de tirer des conclusions robustes au sens physique sur le système physique. Cette approche permet ainsi d'obtenir des résultats qui traduisent mieux les dynamiques de contamination dans les données de contrôle environnemental. L'utilisation de l'indice de similarité par corrélation DTW apporte donc une solution au problème des décalages temporels variés et intermittents.

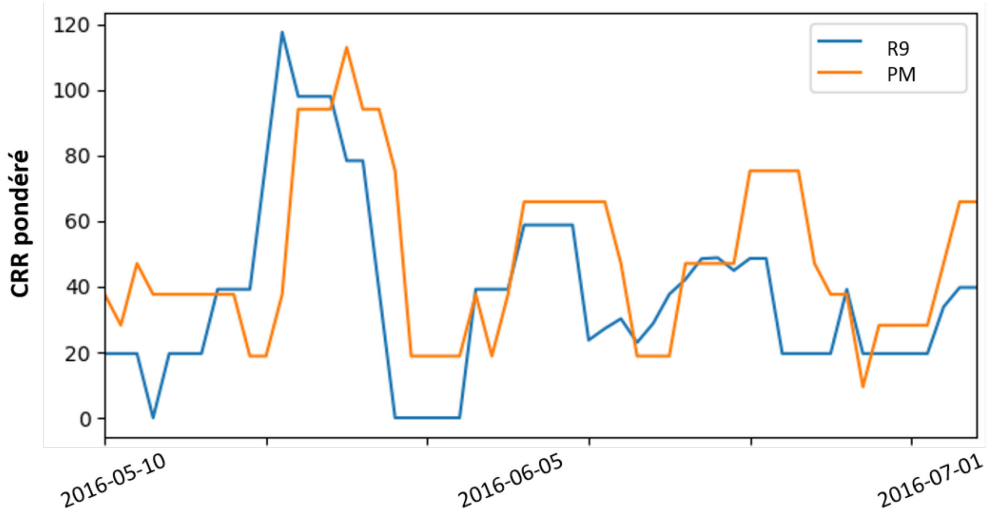

Figure 3.8 : Visualisation des indicateurs de contamination du personnel et de la pièce R9 après l'étape de réalignement

En calculant la corrélation DTW de toutes les paires de variables, une matrice de corrélation DTW a été construite. Le Tableau 3.3 donne les 10 paires de variables avec les corrélations DTW les plus élevées pour les combinaisons avec au moins un indicateur associé aux tests microbiologiques dans une pièce. La matrice complète se trouve en annexe. Le calcul de la matrice de corrélation DTW s'est effectué dans l'environnement python avec la librairie dtw.

Tableau 3.3 : Corrélation DTW pour les 10 paires de variables avec les valeurs les plus élevées

Variable 1	Variable 2	Corrélation DTW	Intervalle de confiance 95%
R4	R10	0.50	[0.44, 0.55]
R6_P	R6	0.46	[0.41, 0.50]
R5_P	R6	0.44	[0.39, 0.49]
R10	R3_P	0.40	[0.36, 0.45]
R1	R6	0.40	[0.34, 0.46]
R3_P	R3	0.39	[0.33, 0.44]
R1	PM	0.38	[0.33, 0.43]
R3	R4_P	0.38	[0.33, 0.43]
PM	R9	0.38	[0.32, 0.44]
R3	R2_P	0.37	[0.33, 0.42]

Les corrélations DTW obtenues avec les données de contrôle environnemental se situent principalement dans une plage de valeur entre -0.15 et 0.5. Plusieurs caractéristiques des données permettent d'expliquer ces valeurs modérées. En effet, la présence de bruit important dans les données ainsi que les relations complexes et parfois intermittentes entre les variables impactent fortement les valeurs de corrélation. De plus, les comportements humains ont une influence importante sur les données avec la contamination des zones de production par le personnel. Ces comportements sont imprévisibles et difficilement capturable ce qui peut expliquer l'obtention d'indices de similarité plus faibles. Une faible corrélation demeure toutefois très utile s'il y a un bénéfice à connaître cette relation et si elle est statistiquement significative.

Une analyse par « bootstrapping » a été utilisée afin de déterminer les intervalles de confiance 95% et de vérifier si les résultats obtenus par corrélation DTW étaient significatifs. Le « bootstrapping » est une technique qui se base sur un rééchantillonnage aléatoire des données afin d'obtenir une nouvelle distribution à partir des données originales pour le calcul d'une métrique. Le rééchantillonnage se fait en choisissant au hasard des d'échantillons dans la distribution originale avec un même échantillon pouvant être sélectionné plusieurs fois. Dans notre cas, la métrique qui a été réévaluée pour les nouvelles distributions est la corrélation DTW. En effectuant un rééchantillonnage aléatoire plusieurs fois (1000 fois dans notre cas), on obtient un intervalle de confiance de

la métrique d'intérêt qui se base sur 1000 nouvelles distributions obtenues à partir des données originales.

Dans le but de faciliter l'interprétation des évaluations de similarité, les éléments de la matrice de corrélation DTW ont été agglomérés avec l'algorithme hiérarchique « single-linkage ». L'algorithme groupe les données en initialisant chaque variable comme un groupe distinct. Durant la première itération, l'algorithme groupe ensemble les deux variables avec l'indice de similarité le plus élevé pour créer un nouveau groupe composé de ces deux variables. Ensuite, lors des itérations subséquentes, la même évaluation est faite afin de créer des groupes englobant de plus en plus de variables. Le regroupement prend fin lorsque toutes les variables se retrouve dans le même groupe. En utilisant cette information pour réorganiser l'ordre des variables, une nouvelle matrice de corrélation DTW agglomérée a été obtenue. Cette matrice est illustrée à la Figure 3.9. Dans cette matrice, les carrés unitaires correspondent à la corrélation DTW entre les variables sur les lignes et les colonnes. Les groupes de carrés verts près de la diagonale représentent les indicateurs de contamination avec des tendances et variations similaires. Cet outil permet ainsi d'extraire les patrons de corrélation dans les données.

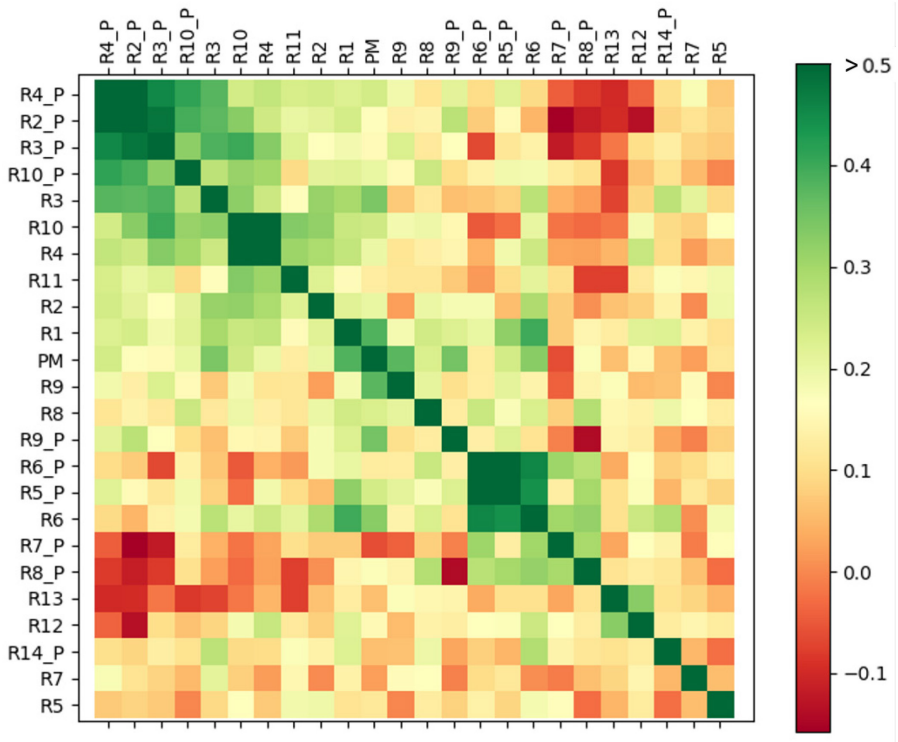

Figure 3.9 : Matrice de corrélation DTW agglomérée

L’interprétation de cette matrice permet de tirer de premières conclusions sommaires sur les tendances dans les données de contrôle environnemental. Par exemple, le groupe de variables dans le coin supérieur gauche de la matrice de corrélation nous indique qu’il y a une corrélation forte pour le nombre de particules dans les pièces R4, R10, R2 et R3. Les pièces R4, R2 en R3 sont des corridors de sorties de la zone aseptiques. Il est donc possible que le personnel génère beaucoup des particules lorsqu’ils retirent leur équipement de protection. Ceci permettrait d’expliquer le regroupement de ces variables dans la matrice. Une autre hypothèse envisageable est que le système d’aération est commun pour tous les corridors de sortie.

Un autre exemple de conclusion intéressante est la corrélation forte entre la pièce R1 et la contamination sur le personnel (PM). En effet, la pièce R1 est utilisée pour enfiler l’équipement de protection alors que les tests sur le personnel sont réalisés lors de leur sortie. Cette information peut donc indiquer que le personnel se contamine pendant qu’ils enfilent leur équipement.

Des investigations plus poussées avec des experts sur le site sont nécessaires pour comprendre en détail les tendances et patrons exposés par la corrélation DTW. Bien que ce premier outil apporte une solution pour l’identification des patrons de contaminations, la visualisation et l’interprétation des résultats demeurent complexes. Comme un des objectifs du projet était de développer des outils de visualisation pour faciliter l’interprétation des résultats, une approche par graphique à nœud a aussi été explorée.

3.4. Graphique à nœuds

Le graphique à nœuds des données de contrôle environnemental se base sur la matrice de corrélation DTW qui donne la similarité de toutes les paires de variables possibles. Le module NetworkX sur python a été utilisé afin de construire un graphique à nœud en se basant sur l'algorithme de force de Fruchterman-Reingold [40]. Cet outil utilise un espace 2D (les distances et orientations n'ont pas de signification), des nœuds (représentant les variables) et des liens (relations fortes) pour visualiser les patrons de relations entre les variables. Un lien est créé entre 2 nœuds lorsque l'intensité de la similarité entre 2 variables est supérieure à un seuil. Ceci se traduit par une corrélation DTW supérieure à une certaine valeur pouvant être choisie par l'utilisateur. Pour les données de contrôle environnemental, la grosseur des nœuds est proportionnelle au nombre de résultats CRR positifs obtenus dans chaque pièce. Ceci permet de facilement distinguer les pièces avec beaucoup de résultats positifs.

En choisissant un seuil bas, beaucoup de relations vont être représentées ce qui peut rendre l'interprétation des résultats difficile. À l'inverse, un seuil trop élevé va conduire à un graphique avec uniquement quelques relations ne traduisant ainsi pas toutes les dynamiques importantes dans les données. Une interface interactive a été programmée avec python afin de pouvoir faciliter le choix du seuil. Pour les données de contrôle environnemental, un seuil sur les percentiles de toutes les corrélations DTW a été utilisé. Les corrélations appartenant au 90^{e} percentile des corrélations DTW ont été conservées.

Le graphique à nœuds obtenu avec les données de contrôle environnemental est présenté à la Figure 3.10. Ce graphique permet de facilement évaluer les relations entre les pièces et d'identifier les dynamiques spatiales présentes dans les données. Bien que l'information comprise dans ce graphique soit similaire à celle de la matrice de corrélation (Figure 3.9), la visualisation sous forme de graphique à nœuds rend l'interprétation des résultats beaucoup plus simple et concise.

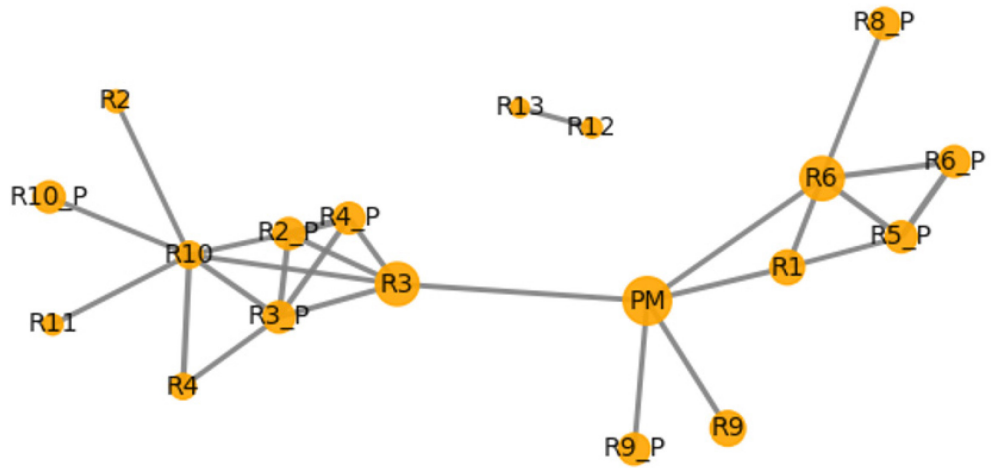

Figure 3.10 : Graphique à nœuds des relations dans les données de contrôle environnemental

Par exemple, la corrélation forte précédemment identifiée entre la contamination du personnel (PM) et la pièce R1 est encore plus facilement repérable. De plus, on constate que la variable PM est aussi fortement corrélée à la pièce R6 qui est elle aussi une zone d'habillement en amont de R1. On peut donc conclure que la contamination du personnel est fortement corrélée à celle des pièces lors de l'enfilement des habits de protection. La mise en évidence de cette relation permet ainsi d'orienter les investigations ainsi que les actions correctives pour l'amélioration du contrôle environnemental.

On peut aussi remarquer qu'il semble y avoir des groupes de pièces avec des patrons de contamination similaires au sein de l'usine. Dans le but de faciliter l'analyse des résultats, ces groupes ont été manuellement colorés et représentés sur le schéma de la carte de l'usine avec la couleur de leur groupe respectif (Figure 3.11).

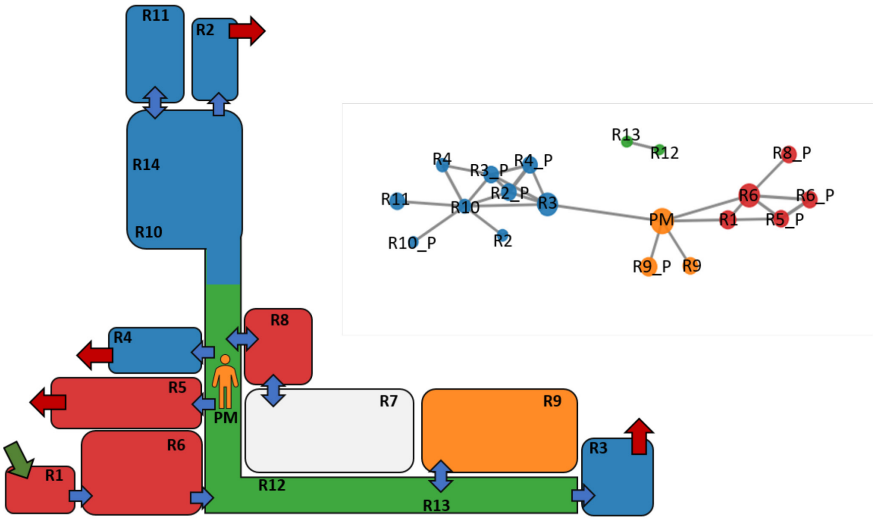

Figure 3.11 : Schéma de la carte de l'usine colorée avec les groupes manuellement identifiés à l'aide du graphique à nœuds.

Cette visualisation des résultats de l'analyse par corrélation DTW permet de mettre en évidence la présence de patrons de contamination pour les pièces proches les unes des autres. En effet pour les groupes de pièces rouges et bleus, la contamination d'une pièce mène souvent à la contamination des pièces environnantes. Ceci est probablement dû aux pratiques de nettoyages ou à la propagation des contaminants dans l'air. La même conclusion peut être tirée du groupe vert qui rassemble les deux zones du même corridor de la zone de production.

Cette visualisation est aussi utile pour faire ressortir des dynamiques particulières qui ne seraient pas attendues au sein des données. Par exemple, le fait que la pièce R3 appartienne au groupe de pièces bleues peut paraître surprenant en raison de sa position spatiale éloignée. Une investigation plus poussée permettrait de comprendre les raisons de cette corrélation forte et d'éventuellement améliorer les pratiques en place.

Une autre conclusion très intéressante qui peut être tirée du graphique à nœuds est par exemple la position centrale de la variable associée à la contamination du personnel (PM). En effet, le nœud PM semble faire le lien entre 3 zones isolées du plancher de production identifiées en rouge, bleu et orange. Ceci est logique car le personnel contribution fortement à l'entrée et la propagation des contaminants dans les zones aseptiques.

Une étude plus approfondie des résultats n'est pas présentée dans ce document, car l'objectif principal était le développement d'une approche d'analyse pour faciliter la

compréhension des données spatio-temporelles. L'outil de visualisation vient répondre à cet objectif avec des résultats simples et facilement interprétables. L'identification et la présentation des tendances et patrons de corrélation dans les données apportent ainsi des pistes sérieuses d'investigation pour comprendre et améliorer le contrôle environnemental.

3.5. Conclusion sur l'analyse par corrélation DTW

L'analyse par corrélation DTW et les outils de visualisation tels que la matrice de corrélation, le graphique à nœuds et la corrélation roulante apportent une solution pour l'étude des données de contrôle environnemental. Cette approche permet de faire ressortir les patrons de corrélations dans les données et de visualiser les résultats de façon à faciliter leur interprétation.

Cependant, l'analyse se basant sur la corrélation DTW a aussi des limites. En effet, le graphique à nœuds permet uniquement de visualiser les patrons de corrélation spatiale entre les pièces. Bien que l'outil de corrélation roulante permette d'étudier plus en profondeur l'évolution de la similarité entre 2 variables dans le temps, cette approche ne permet pas d'avoir une vision globale temporelle des données de contrôle environnemental.

De plus, l'analyse de nouvelles données dans une optique de contrôle en continu est assez complexe. En effet, une quantité importante de données est nécessaire afin de construire un nouveau graphique à nœuds ce qui rend difficile l'utilisation de l'outil sur une base hebdomadaire ou mensuelle par exemple.

Finalement, l'approche par corrélation DTW se base sur l'évaluation de similarités par paires de variables. Bien que le graphique à nœud permette d'avoir un vison global de toutes les relations importantes dans les données, le calcul des similarités ne se fait pas en considérant toutes les variables.

Une approche multivariée pour l'analyse des données de contrôle environnemental a donc été développée. Le chapitre suivant décrit en détail la méthodologie, des applications possibles ainsi que l'étude de cas pour les données de contrôle environnemental.

4. RÉSOLUTION MULTIVARIÉE DE COURBE

Auteurs et affiliation :
A. Vielfaure : étudiant à la maîtrise, Université de Sherbrooke, Faculté de génie, Département de génie chimique et de génie biotechnologique.
A. Cournoyer: Manager Senior, Pfizer Canada, PMAC
R. Gosselin : Professeur, Université de Sherbrooke, Faculté de génie, Département de génie chimique et de génie biotechnologique.

Date de soumission : Soumission le 25 mai 2020, accepté en date du 31 aout 2020

Revue : Industrial & Engineering Chemistry Research

Titre français : Identification de dynamiques et patrons à partir de données spatio-temporelles bruitées avec la résolution multivariée de courbe

Contribution au document :
Cet article présente une contribution importante au projet de recherche. Il contribue en élaborant sur l'analyse multivariée de données spatio-temporelles bruitées avec l'algorithme multivarié de courbe (MCR). Plus spécifiquement, l'article présente une nouvelle approche pour l'étude de ce type de données en utilisant la variante MCR-LLM développée pour des images spectrales bruitées.

En démontrant les performances supérieures de l'algorithme MCR-LLM à l'algorithme MCR standard par régressions alternées (ALS), l'article apporte une solution au problème de l'analyse de données spatio-temporelles à faible rapport signal sur bruit.

Une comparaison quantitative et qualitative entre les deux variantes de la méthode MCR (MCR-ALS et MCR-LLM) sur deux jeux de données avec des dynamiques et patrons connus a permis de confirmer la pertinence d'utiliser l'analyse multivariée par MCR-LLM pour l'étude des données de contrôle environnemental.

L'article présente finalement l'application de l'algorithme ainsi qu'une analyse des résultats pour les données du partenaire industriel. Des outils de visualisation facilitant l'interprétation des résultats sont aussi présentés à des fins de réplication pour d'autres données du partenaire industriel.

Résumé français :
Cet article présente une nouvelle approche pour l'étude de données spatio-temporelles bruitées à l'aide de l'algorithme de réduction de dimension par MCR-LLM. Précédemment développé pour l'étude d'images spectrales bruitées, l'algorithme utilise une étape d'optimisation plutôt que des régressions linéaires alternées afin de produire des résultats plus robustes et faisant plus de sens en présence de bruit. L'algorithme standard MCR-ALS et la variante MCR-LLM ont été utilisés pour l'étude de trois jeux de données différentes dans le but d'évaluer leur performance.

Les deux algorithmes ont d'abord été utilisés sur un ensemble d'images de chiffres écrit à la main et corrompu de façon synthétique avec différents niveaux de bruits. Les performances de MCR-LLM ont quantitativement surpassés celles de MCR-ALS pour tous les niveaux de bruit testés. La variante LLM a été en mesure de mieux extraire l'information pertinente des données, et ce même en présence d'un bruit élevé.

Des données météorologiques pour plusieurs stations américaines durant la saison des ouragans ont ensuite été analysées avec les deux variantes de MCR. L'analyse avec MCR-LLM a surpassé celle par MCR-ALS en produisant des résultats permettant d'identifier trois conditions météorologiques typiques, dont une associée aux passages d'ouragans. La variante ALS a quant à elle eu beaucoup de difficulté à produire des résultats interprétables.

MCR-LLM a finalement été utilisé pour analyser des données spatio-temporelles provenant d'un programme de contrôle environnemental en milieu pharmaceutique. L'algorithme a permis de mettre en évidence des patrons spatiaux de contamination et de grandement faciliter la visualisation des dynamiques temporelles dans les données.

La méthodologie présentée dans cet article peut être répliquée afin d'analyser d'autres données spatio-temporelles.

Extracting meaningful patterns from noisy spatiotemporal datasets with Multivariate Curve Resolution

4.1. Abstract

This paper presents a novel approach to dealing with noisy spatiotemporal datasets with a data reduction algorithm called MCR by Log Likelihood Maximization (MCR-LLM). Previously applied to spectral data, MCR-LLM uses an optimization step instead of traditional alternating least square (ALS) regression to produce more meaningful results when dealing with noisy data. MCR-LLM and MCR-ALS were used on three different datasets to assess their performance.

The first consisted in a set of handwritten digit images synthetically corrupted with variable and controlled levels of noise. MCR-LLM was shown to quantitatively outperform MCR-ALS for all levels of noise while maintaining meaningful results even with very noisy data.

The second dataset consisted in spatiotemporal meteorological data from different American weather stations during the hurricane season. The LLM variant outperformed MCR-ALS by producing meaningful results that highlighted typical types of weather conditions and a component associated to hurricanes while MCR-ALS struggled to produce interpretable results.

MCR-LLM was then used to analyze a spatiotemporal dataset from a pharmaceutical environmental monitoring program. The algorithm highlighted clear spatial patterns and greatly facilitated visualization of temporal contamination dynamics.

The methodology used in this work could be applied to other noisy spatiotemporal datasets.

4.2. Introduction

Industrial process monitoring is becoming increasingly complex with the recent explosion of data collection technologies. Very large amounts of data are now being generated allowing real-time in-depth analysis. Environmental monitoring programs in the food and pharmaceutical industries are faced with this challenge and more advanced data analytic techniques are key in ensuring increased product quality. In the case of aseptic products, this implies ensuring drug sterility. Environmental monitoring activities are generating huge spatiotemporal datasets from microbial tests and swabs, but they often use data analysis tools that are inappropriate for a thorough data understanding. Current methods are generally very limited in terms of data understanding and usually based on direct data observations with control charts [7][41]. Some works have proposed using more complex charts with frequency models for better interpretation [9][6], but no attempt at multivariate spatiotemporal analysis of pharmaceutical environmental monitoring data has been made so far.

Most microbial results are reported in terms of colony-forming units (CFU) on culture plates which is an indicator of the level of contamination based on the number of cells that give rise to colonies after a predetermined incubation period. The rare and sporadic nature of contamination outbreaks is a major challenge for environmental monitoring data analysis. With very few high CFU counts, real contamination trends (signal) can be hard to differentiate from noise associated with normal operating conditions or plate mishandling. Additionally, even though culture plates are usually representative of the worst-case scenario, null CFU counts do not guarantee the absence of microorganisms in the manufacturing environment as sampling was only performed on a small specified area. Signal-to-noise ratio (SNR) is therefore usually very low with this data type.

Popular approaches for the investigation and understanding of multivariate datasets include data reduction techniques. Latent variable algorithms that seek to explain as much variance as possible with a reduced number of components can extract patterns in the data and facilitate interpretation. MCR-ALS (Multivariate Curve Resolution) [21] is one such algorithm that uses Alternating Least Square regressions to extract meaningful components. The algorithm has been successfully used to analyze spatiotemporal datasets for water pollution pattern identification [25][26][27] and air pollution analysis in urban areas [28]. However, the issue of inherent noise in the data has only been briefly

addressed with non-spectral MCR analysis and accurate prior noise estimation is still necessary [42][43]. The regression steps of Alternating Least Square cannot cope with low signal to noise ratio datasets, which can greatly impact results and interpretation [29].

MCR-LLM (Log Likelihood Maximization) has recently been presented as an alternative to MCR-ALS for dealing with low count noisy spectral images [29]. The variant uses an optimization step instead of multilinear regressions to extract components and was shown to outperform MCR-ALS for different spectral datasets [30][31].

The aim of this study was to compare the performances of MCR-ALS and MCR-LLM for noisy non-spectral datasets. In this manuscript, we briefly recall the background of both methods and then compare the performances of both algorithms in three situations. First, with a simple dataset of 2D images with controlled levels of noise: images of hand-written digits. Second, the algorithms were tested on a spatiotemporal dataset in which major events perturbed noisy measurements: meteorological data from multiple stations during the hurricane season. Third, another spatiotemporal dataset was used in which relatively minor variations perturbed noisy measurements: environmental monitoring data from a pharmaceutical manufacturing site, representing a real case where sparse data are available and where MCR-LLM can be successfully used.

4.3. Methods

4.3.1. MCR-ALS

MCR-ALS performs dimensionality reduction with a linear model that estimates a data matrix D (M × N) with a reduced contribution matrix C (M × K) and loadings matrix S (N × K) [21]. The linear transformation can be written as:

$$D = C S^T + E \tag{4.1}$$

where E (M × N) is the residual matrix representing the error. The algorithm is based on matrix factorization but differs by introducing meaningful constraints during the multilinear regression step [22], [23]. Constraints such as non-negativity, closure, unimodality or symmetry can force the shape of the components in the contribution matrix C and loadings in S. Such constraints can increase the robustness of the regression steps and allow for easier interpretation of the results with physically meaningful loading and contribution profiles.

Additionally, the non-nested nature of MCR-ALS makes the model more sensitive to the number of components K chosen for the dimensionality reduction. Conceptually, this means that the calculation of subsequent components will alter the previously calculated contributions and loadings. It is therefore important to correctly chose the number of components K to perform the dimensionality reduction in order to get interpretable results. Techniques such as effective rank analysis [24] can help determining how many components to use based on the data.

4.3.2. MCR-LLM

The MCR-LLM variant [29] is similar to MCR-ALS but considers the inherent noise in the data while calculating C. With noisy data, multilinear regressions between the data matrix D and S using the ALS algorithm can lead to significant errors which are meaningfully corrected by the application of constraints. These errors often prevent convergence or lead to incorrect loading and contribution matrices. Instead of using a multilinear regression between S and D, MCR-LLM uses likelihood maximization based on signal noise [44]. For each observation in the data matrix, a likelihood distribution L_m is calculated

and then maximized to give the contribution values for all components. By repeating this process for every data point in D, the contribution matrix C can be calculated.

A scaling step of the C matrix is also performed before calculating S with multilinear regression. This is done in order to avoid abrupt changes of S that might be due to important noise characterizing the data, therefore maintaining meaningful profiles.

Further details on both MCR-ALS and MCR-LLM can be found in previous works [29][21][45].

4.3.3. Spatiotemporal data organization

Spatiotemporal datasets are characterized by three dimensions; the time of measurement, the variables measured and the location of the measurement. In order to apply MCR algorithms, a reduction to two dimensions is necessary. One possible data transformation method used in this work uses the spatial information to unfold the 3D data matrix ($\underline{D}$) into a stacked matrix D_{unf} [25]. This way of organizing the data introduces a new trilinear constraint which simplifies the interpretation of the dataset by forcing the different locations to share common loadings in the S matrix. This way of reorganizing 3D data is especially useful in cases where different locations are expected to share similar patterns.

MCR algorithms have two types of ambiguities that make identification of a unique solution difficult [46]. Intensity ambiguities are characterized by different scaling factors of the contributions and loadings matrices. Applying closure or non-negativity constraints can solve this issue but does not always guarantee a unique solution. Such cases are characterized by an additional rotational ambiguity which is usually the most difficult to deal with [47]. By adding the trilinear constraint with the stacked data matrix, the rotational ambiguity can be solved and a unique solution can be found [48].

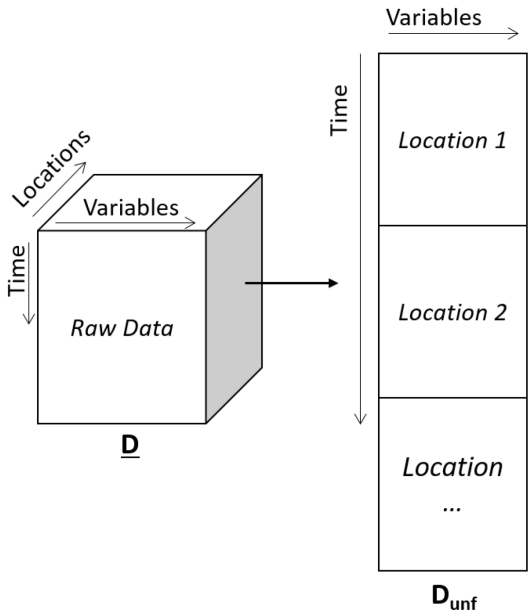

Figure 4.1 : Spatiotemporal data organization for MCR decomposition

4.4. Datasets

4.4.1. Digits dataset:

Characteristics :

The digits dataset from the UCI repository [49] is a set of 1797 handwritten digits (i.e. numbers 0 to 9). Each sample is written onto an 8 × 8 grid, or image, with pixel intensity values ranging from 0 to 16.

Pre-processing :

To apply the MCR algorithms on the digits, a simple unfolding step was performed to transform the 2D images into 1D vectors with 64 variables. The original data was then corrupted with different levels of Poisson noise in order to assess its impact on the performance of both algorithms (ALS vs LLM). Since Poisson noise is signal dependent, the equation below was used to regulate the amount of noise added.

$$D_{noise} = \frac{Poisson(\frac{D_{or}}{s_{max}} \times n)}{n} \times s_{max} \tag{4.2}$$

where *n* is the scaling factor inversely proportional to the amount of noise added, s_{max} is the maximum pixel intensity and D_{or} the raw data matrix.

As pixel intensity values varied from 0 to 16, s_{max} was therefore set to 16. With a very high *n* parameter, the discrete Poisson distribution can correctly approximate the data with only integers whereas a small value for *n* will lead to poor approximations. Figure 4.2 shows the impact of noise depending on the *n* value.

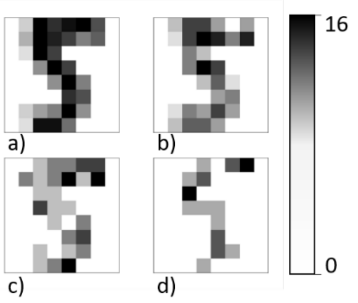

Figure 4.2 : Impact of different n values (∞,5,2,1 respectively) on a normalized digit image

Performance index :

The performance index *P* used to quantitively assess the performance of the MCR algorithms was first proposed by Lavoie et al. [29]. The idea is to use the loading profiles extracted with the raw dataset (no noise) as references and compare them to the new loading profiles extracted with different levels of noise. The equation used is described below:

$$P = 1 - \frac{\sum_{n=1}^{N}\sum_{k=1}^{K}\left|Rc_{k,n} - Sc_{k,n}\right|}{K} \tag{4.3}$$

Where *Rc* is the matrix containing the reference profiles (from the MCR decomposition without noise), *Sc* the matrix with the noisy profiles and K and N being the number of loadings and the number of variables (pixels) respectively.

4.4.2. Hurricane dataset:

Characteristics :

The hurricane dataset was obtained online [50] and consists of meteorological data from 23 different American weather stations over a two-month period ranging from the beginning of September 2018 to the end of October 2018. The geographical locations of the stations are shown in Figure 4.3.

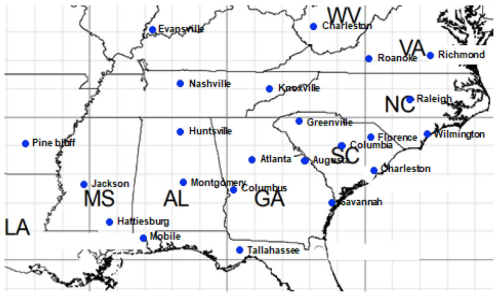

Figure 4.3 : Spatial distribution of the different meteorological stations

For each station, 10 different variables were recorded at an hourly rate for a total of 1488 hours over 62 days. These variables are: 1) dew point temperature, 2) dry temperature, 3) humidity, 4) precipitation, 5) sea level pressure, 6) station pressure, 7) visibility, 8) wet-bulb temperature, 9) wind direction and 10) wind speed. All variables were kept for the analysis. The temporal range was specifically chosen for the presence of hurricanes passing through the United States during the September and October months. Hurricanes Michael and Florence are the two major hurricanes that affected many of these weather stations during this period. The raw data was therefore a 3D spatiotemporal matrix measuring ($1488 \times 23 \times 10$) which corresponds to a total amount of 342 240 individual values.

Pre-processing :

The dataset contained less than 1.6% missing values and these values were filled by linear interpolation. The moving average over a period of six hours was then applied for each variable to incorporate temporal dynamics at each time point. Variables were scaled to unit variance to consider the different units and scales of the meteorological variables. The individual 2D matrices for each station were stacked location-wise to produce the

stacked data matrix D_{unf} presented in Section 2. By applying this data transformation, all stations were forced to share a common data decomposition with the MCR algorithms.

4.4.3. Industrial Environmental Monitoring dataset

Characteristics :

The environmental monitoring dataset used in this work comes from a pharmaceutical manufacturing site which collected microbial samples in their aseptic manufacturing site over a period of three years. Multiple microbial analyses with different units and scales were reported daily at various locations throughout the site and stored. Such microbial analyses include plate colony counting from swabs on personnel, factory floors and equipment, non-viable particle counts for the air, and settle plate colony counting for passive air testing. The aseptic area of the manufacturing site is also characterized by multiple areas with different pharmaceutical grades that require different sampling frequencies. Additionally, changes in aseptic practices has led to increased environmental control which translates into a lot of null results (no microbial detection) during the environmental analysis. This also means that detection of contamination is much harder and therefore much more impacted by noise (false positives). Some studies have shown that the use of traditional quantitative evaluation for this type of data is inappropriate and approaches that use the frequency of positive recoveries as an indicator of the level of contamination are better suited [6][51]. The raw data consisted of a 3D matrix with daily positive recovery counts of multiple microbial analyses and quantitative particle counts taken in various locations throughout the aseptic area over a period of three years ranging from 2015 to 2018. In total, 15 different locations were sampled at least daily for 988 days with five different microbial analysis types. Since some analyses were only performed in specific locations, the raw 3D matrix ($988\times 15\times 5$) was interspersed with variables without any values for some locations.

Pre-Processing :

Different data pre-processing steps were tested for the environmental monitoring dataset. Due to the rare nature of contamination outbreaks and low frequency of high CFU counts, all the different microbial analyses reported with a CFU (e.g. floor swabs, equipment swabs, passive air samples) were grouped together under a more representative overall

microbial level indicator for each location. This new variable expressed the daily contamination in terms of number of positive recoveries (CFU counts higher than 0) per location. This way of pre-processing the data was also a way of dealing with the sparse nature of the spatiotemporal dataset. The variables associated to quantitative inert particle counts per location were kept separately because of the different unit used. The second pre-processing step was to apply a moving sum over a seven-day period for every location to better capture contamination trends. Both indicators were then scaled to unit variance for each location to consider the different units between particle counts and number of positive recoveries. Different weights were then added to the various locations to mirror testing frequency. This was done to prevent areas with few analyses and therefore few detected excursions to be overweighed by the scaling step. The weights were calculated with the cube root of the total number of analyses performed per locations. Such a scaling factor is arbitrary and must be adapted, or removed, for a given environmental monitoring dataset. Preliminary analyses showed that using such a scaling increased the weight of the rooms that had undergone significant testing while lowering the weight of the rooms that had undergone very little. Given the low SNR of the dataset, rooms that had undergone very little testing were deemed to be less reliable, thus justify the use of the scaling technique. The dataset contained 3.8% missing values that were filled by using a PCA model with five components. The 3D data matrix was stacked variable wise to produce a 2D matrix D_{unf} (988×24) of 24 variables representing the daily microbial level and number of particles for various locations for a total of 988 days.

4.5. Results

4.5.1. Digits dataset:

Dimensionality reduction with both MCR-ALS and MCR-LLM was initially performed using 13 components based on the calculated effective rank of 13.1. The expected data in the loading matrix S was the average shape of the different digits in the dataset (i.e. digits 0 to 9). While an effective rank higher than 10 may at first may appear surprising, it is important to recall that the dataset consists of handwritten numbers and that multiple variants are common for some digits. Figure 4.4 shows the reconstructed loading profiles from the MCR-LLM reduction with 13 and 11 components when transforming the 64-element loading vectors back to 8×8 image matrices. The algorithm was able to extract loadings that clearly represented the average shape of each individual digit (or different ways of writing a digit) in the dataset. The contribution matrix C represents the relative importance of each component in explaining the shape of digits in the image dataset. High contribution values for the first component would translate in a digit with a shape similar to the first extracted component in the loading matrix. To achieve these results, non-negativity and closure constraints were applied. Non-negativity was used to ensure that the contribution matrix exhibited only positive values while the closure constraint was used as a normalization constraint that forces, for each image, the sum of all contribution values to 1.

We can see in Figure 4.4 that with 13 components there is a repetition of the digits ‘1’, ‘5’ and ‘7’. We then only used 10 components (for the 10 different numbers in the dataset) and noticed that the algorithm extracted the two ways of writing the digit ‘1’ before extracting a loading profile for digit ‘8’. Such a decomposition is perfectly valid, even if it does not correspond to our expectations. It simply underlines the large variability in the ways of writing digit ‘1’. By using 11 components (profiles are shown in Figure 4.4b) we were then able to get reconstructed profiles for every digit in the dataset with a single repetition of the digit ‘1’. The remaining of the analysis was therefore done with 11 components.

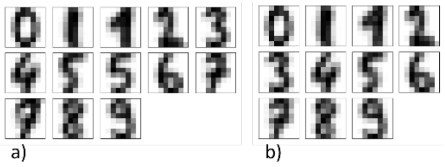

Figure 4.4 : Reconstructed loading profiles for MCR-LLM with 13 (a) and 11 (b) components

Comparing loading profiles :

Figure 4.5 illustrates the reconstructed loading profiles from MCR-LLM (a, c) and MCR-ALS (b, d) for the digits dataset with 11 components. The reference loadings in (a, b) were extracted with the raw dataset whereas loadings shown in (c, d) are from a noisy dataset with a parameter value of n=2. All loadings were normalized prior to comparison. For the raw dataset, both methods could extract loadings profiles that were very similar and easily interpretable. We can clearly identify the shapes of all 10 digits in the dataset with two ways of writing the digit ‘1’ in both cases. Loadings extracted with MCR-LLM appeared to have a better contrast with the background. MCR-ALS produced loadings with very high values in specific positions whereas MCR-LLM distributed the intensity more evenly over the whole digit shape.

With strong noise (Figure 4.5d), MCR-ALS failed to extract meaningful profiles. The digit shapes were no longer identifiable. The limitations of MCR-ALS for noisy datasets were very clear when compared to the loading profiles from MCR-LLM (Figure 4.5c). The loadings extracted by MCR-LLM for the noisy dataset remained very similar to the reference profiles with small changes in the distribution of intensities. The main difference was the loading profile representing digit ‘8’ that was switched for a second way of writing digit ‘7’. This was most probably due to the difficulty of the task of identifying the digit ‘8’ (which can resemble digit ‘1’) when high levels of noise are present.

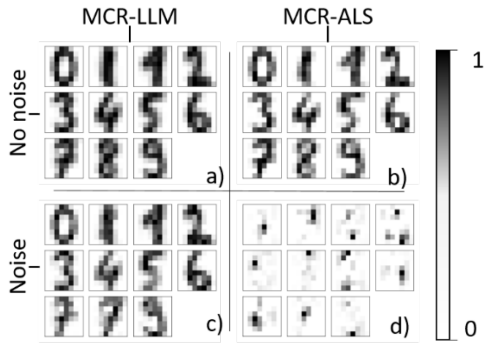

Figure 4.5 : Reconstructed loading profiles from MCR-LLM (a,c) and MCR-ALS (b,d) without noise (a,b) and with a strong noise n = 2 (c,d)

Performance assessment :

The performance index *P* was used to quantitively assess the differences between MCR-ALS and MCR-LLM data decomposition with noisy datasets. Ten different noise levels were used ($1<n<50$), to create 10 new noisy digits datasets. The performance values were then calculated by comparing the reference loadings from the raw data with the loadings from the noisy datasets. Values near 1 denote strong similarities between the references and computed loading profiles.

MCR-LMM outperformed MCR-ALS for all datasets tested (Figure 4.6). The SNR was calculated with the square root of the *n* parameter because the noise was added with a Poisson function with a mean of *n*. For SNR>4, both algorithms behaved similarly and showed strong performances. They were able to extract loading profiles that were very similar to the reference loadings.

Starting at SNR=3, the two algorithms started to significantly diverge, with MCR-LLM's performance remaining high and MCR-ALS's performance dropping considerably. This behavior was observed until around SNR=1.4, where MCR-LLM's performance also started to drop. For very low signal/noise ratios, even the MCR-LLM algorithm's performance suffered and extracting meaningful profiles was difficult.

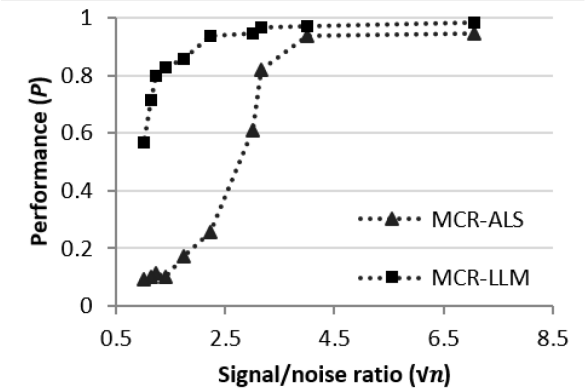

Figure 4.6 : Performance index calculated for MCR-LLM and MCR-ALS with different levels of noise

MCR-LLM was shown to quantitatively outperform MCR-ALS for the digits dataset in the presence of noise. With the performance index used for this dataset, it is possible to demonstrate that the limitations of MCR-ALS with noisy spectral data can also be observed with non-spectral datasets. The LLM algorithm was still able to extract meaningful loadings even with very noisy data. By demonstrating this behavior on a simple and easily interpretable non-spectral dataset, the algorithm can then be used on more complex spatiotemporal data for analysis.

4.5.2. Hurricane dataset:

Data decomposition was performed on the location-wise stacked data matrix D_{unf} with three components based on the calculated effective rank of 3.0. The expected output in the loading matrix S was three loading profiles representing the different typical weather conditions found during the two-month period. This idea was to use the MCR algorithms to extract three different typical weather conditions associated to different relative importance of the various meteorological variables in the dataset. The contribution matrix C represents the relative importance of the different components in explaining the weather conditions of each day. High contribution values for the first component would translate in a day with weather conditions similar to the first extracted type of typical weather condition. Non-negativity and closure constraints were used for both MCR-LLM and MCR-ALS. Non-negativity was used to ensure that the contribution matrix could only have values of 0 and higher. The closure constraint was used to normalize contribution values and have them sum up to 1 for every day of the spatiotemporal dataset. For visualization purposes, we will illustrate the contribution profiles of the Raleigh (North Carolina) station as it is both representative of the broader dataset and clearly illustrates all the points that are to be discussed. The simultaneous analysis for all stations will be presented in Section 4.2.3.

Loadings and contribution profiles :

Results for MCR-LLM and MCR-ALS are presented in Figure 4.7 and Figure 4.8 which show the contributions and loading profiles for all three components. In Figure 4.7, the magnitude of a contribution profile represents how similar each day is to the three typical extracted weather conditions types. For loading profiles (Figure 4.8), the relative magnitude of a variable across all three components represents its importance when associating each day to a typical weather condition type.

For MCR-LLM, the first extracted contribution profile (green component) showed strong expression during the first 45 days and then very low values for the remaining 15 days (Figure 4.7). The first loading profile (Figure 4.8) exhibited higher relative values for temperature variables (Dew point temperature, Dry temperature, Wet temperature), medium relative values for pressure variables (Sea pressure and Station pressure) and lower values for wind variables (Wind direction and wind speed) compared to the other two components. Overall, the loading profile extracted warmer weather conditions with

very low precipitations and medium visibility, which are usually associated with the summer period.

The second MCR-LLM contribution profile (blue) exhibited strong expression mainly when the first profile (green) was weak. A clear duality seemed present between the two components. The second loading profile showed some similarities with the first one with major differences for temperature and pressure variables. The profile extracted colder weather conditions associated with the autumn period. When denser colder air replaces warm air, the measured surface pressure increases explaining the higher relative pressure values for the second loading. MCR-LLM's interpretation of the first two loadings that extracted weather conditions associated to the two seasons was also confirmed by the contribution profiles with high expression transitioning from the first to the second component around mid-October.

The third contribution profile (red) was briefly very strong around the 15th of September and 11th and 25th of October. Those dates correspond with hurricanes Florence and Michael as well as a strong storm. For MCR-LLM, the third loading extracted weather conditions associated with storms and hurricanes. Humidity, precipitation and wind variables exhibited strong relative values compared to components 1 and 2. Additionally, the pressure variables were very low. Overall, the third loading profile described hurricanes and storms very well, displaying high humidity, low pressures, precipitations and strong winds.

For MCR-ALS, the first contribution profile did not exhibit a clear temporal pattern. The profile showed values around 0.5 for the first 45 days and then low values with some abrupt peaks during the last 15 days. The first MCR-ALS loading showed some similarities with MCR-LLM with stronger temperature variables and medium relative pressure importance compared to the other two components. However, the first loading for the ALS variant also exhibited strong relative importance for the precipitation variable and very low values for visibility which complicated result interpretability.

The second contribution profile extracted with MCR-ALS showed relatively smaller values for the first 45 days and higher values for the last 15 days. The observed duality between the first and second component for MCR-LLM decomposition was not as clear for MCR-ALS. Both the first and second components exhibited similar contribution intensities during the first 45 days. The second loading profile extracted with MCR-ALS was very similar to

MCR-LLM's second loading. The main difference was with the visibility variable that exhibited weaker relative values compared to the third component and with the wind variables that showed smaller relative importance compared to the first component.

MCR-ALS's third contribution profile exhibited very low values for all days and showed no interpretable patterns. The third loading exhibited very strong values for visibility but was not significantly associated to any time-periods in the contribution profile. The algorithm was not able to extract a third meaningful component, and contrary to MCR-LLM, the ALS algorithm appeared to mix the hurricanes periods with the first extracted components.

MCR-LLM extracted three meaningful and easily interpretable components: two seasons (green and blue) and a storm component (red) mostly associated to hurricanes. MCR-ALS did not achieve similar results and interpretation was much more difficult. The biggest differences between MCR-LLM and MCR-ALS were observed for the third component (red) which seemed to only capture noise and did not allow for extraction of a separate component associated to hurricanes and storms. Those results highlighted the failure of MCR-ALS to extract meaningful profiles.

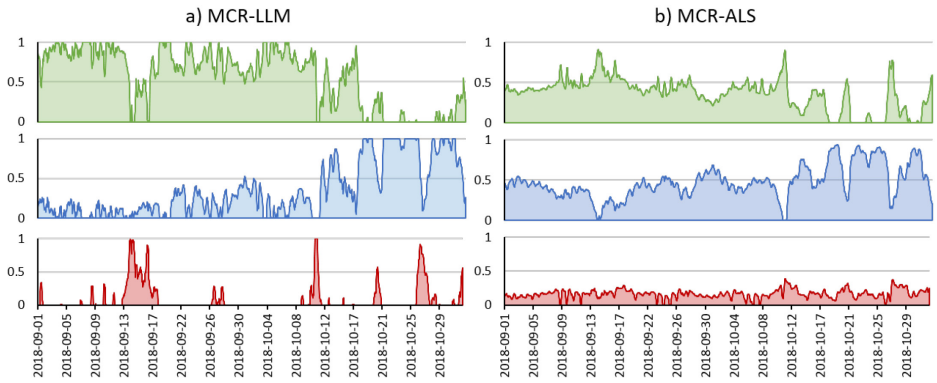

Figure 4.7 : Contribution profiles from three component MCR decomposition for the Raleigh station extracted with MCR-LLM (a) and MCR-ALS (b)

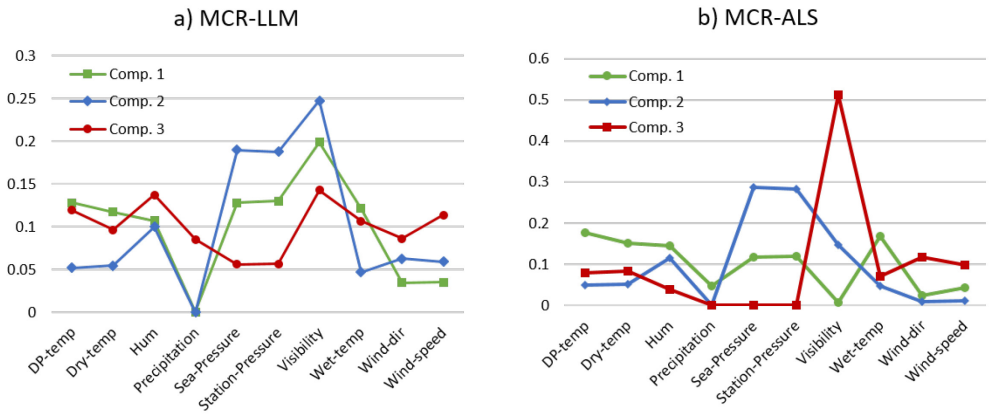

Figure 4.8 : Loading profiles from three components MCR decomposition for all stations extracted with MCR-LLM (a) and MCR-ALS (b)

Frequency analysis :

A frequency analysis was also performed on the MCR contribution profiles (Figure 4.9) to understand the small periodic variations that were observed for both components 1 and 2. The fast Fourier transforms (Figure 4.9) showed a strong frequency at around 1/day for components 1 and 2. The presence of 24-hour variations in the profiles was clearly due to the night/day temperature variations.

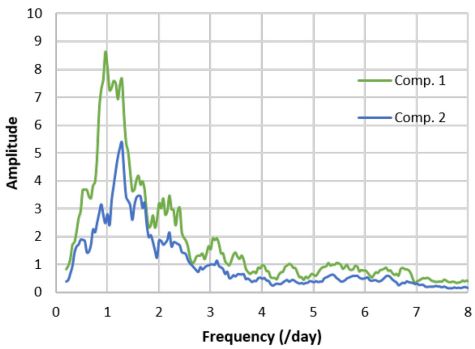

Figure 4.9 : Fast Fourier transform of the first and second contribution profiles from the MCR-LLM decomposition

Discussion :

The results from the MCR-LLM data decomposition in the contribution matrix for all stations were used simultaneously to map the spatiotemporal dynamics of both the season

and hurricane components on a map of the US (Figure 4.10). First, the second component (associated to colder periods) was isolated for all locations and the contribution profiles were used to determine the timepoints at which each location exhibited six consecutive values higher than 0.9. The goal was to determine, for every location, when the transition from summer to autumn was taking place. Figure 4.10 shows the color-coded days of arrival of autumn for each station. A clear north-to-south pattern was identifiable. This pattern, in alignment with the initial hypothesis (autumn with colder weather conditions arriving later in the south) confirmed that MCR-LLM was able to extract meaningful seasonal profiles.

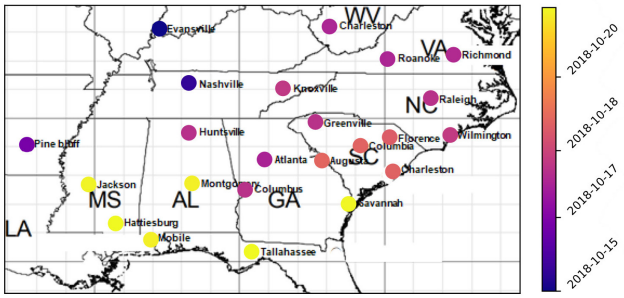

Figure 4.10 : Spatial distribution of autumn arrival for every station colored based on the time of arrival

The same methodology was used with the third contribution profile (storm component) to track hurricane Michael from October 10th to 12th. Figure 4.11 shows the color-coded stations according to the time at which the contributions values for the storm components were at their highest. Additionally, the size of the dots is proportional to the intensity of the contribution profile C meaning that stations directly in the hurricane's path will be larger. A trajectory was also calculated by averaging the weighted coordinates of every station based on the intensity of the storm contribution profiles at each time point. The calculated trajectory shown with the colored line was found to be very similar, spatially and temporally, to the real trajectory represented by the faded black line. The divergence between the calculated and real trajectories at the beginning of the hurricane's path was most probably due to the unevenly distributed locations of the meteorological stations. For the first 12 hours of the trajectory, only one station is located to the east of the real path (Tallahassee) whereas multiple stations are located to its west. When calculating the weighted averaged coordinates of every station based on the intensity of the storm contribution profiles, the results were biased toward the area with more stations. This

explains the observed shift to the west at the beginning of the calculated trajectory. With more evenly distributed stations, we do not expect this behavior to be present.

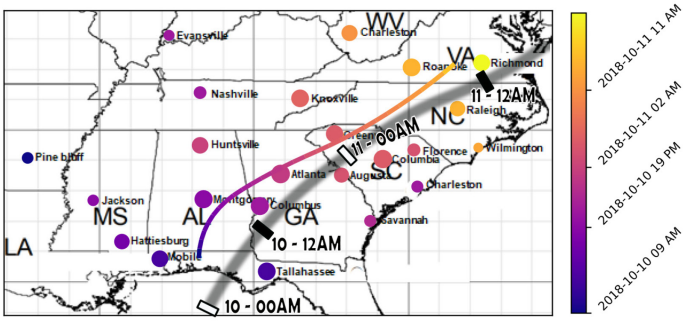

Figure 4.11 : Color-coded spatial and temporal distribution of the storm component with the colored calculated trajectory and the real trajectory in shades of black. The black and white rectangles on the real trajectory represent the exact location of the hurricane's eye at 12-hour intervals

Overall, the hurricane dataset highlighted the shortcomings of MCR-ALS in dealing with noisy datasets. MCR-LLM could extract meaningful loading and contribution profiles with coherent spatiotemporal dynamics that allowed for pattern identification and multivariate analysis of noisy spatiotemporal data.

4.5.3. Environmental Monitoring dataset

Performance comparison between MCR-ALS and MCR-LLM with the environmental monitoring dataset was more complicated and less relevant because the expected output was unknown. Based on the better performances of MCR-LLM for both the digits and hurricane datasets, the industrial environmental monitoring (EM) dataset was analyzed only with the LLM variant to assess if the algorithm could produce intelligible results. The calculated effective rank for the EM dataset was 5.8. However, we opted to perform multiple MCR-LLM models with an increasing number of components. Using a simple model with only two components, we hypothesized that the model would distinguish between problematic periods marked by contamination outbreaks and periods of normal operating conditions. Following this hypothesis, adding more components would then divide problematic periods into subtypes marked by contamination outbreaks associated with different microbial variables and locations.

The expected output in the loading matrix S was the different operating conditions identified during the three-year period in the dataset. The loading profiles would contain the relative importance of each variable (i.e. microbial level and particle counts for every location) to discriminate between normal operating conditions and the different possible contamination patterns. The contribution matrix C represents the relative importance of the different components in explaining the operating conditions for different time periods. High contribution values for the normal operating conditions component would imply a day without any significant contamination whereas high values for the second component would mean a contamination trend associated to locations and microbial variables with high importance for the second component. Non-negativity was used to ensure that the contribution matrix could only have values of 0 and higher. The closure constraint was used to normalize contribution values and have them sum up to 1 for every day of the spatiotemporal dataset.

To assess the ability of MCR-LLM to extract a normal operating condition component, we calculated the Pearson correlation between the sum of all microbial recoveries for every location and the contribution profile identified as the normal conditions. An initial correlation coefficient of 0.66 with two components suggested that the algorithm was able to correctly extract components associated to normal and contaminated periods.

Increasing the number of components allowed to discriminate between different contamination patterns while maintaining the normal behavior contribution profile. To assess the robustness of MCR-LLM data decomposition with different numbers of components, cross-validation was performed by randomly removing 30% of the data in two-week blocks and then calculating the similarity between the new loading matrix S and the reference loading matrix with eq 1 (performance index *P*). The cross-validation results as well as the Pearson correlation between the identified normal behavior component and the sum of all microbial recoveries are shown in Tableau 4.1.

Tableau 4.1 : Correlation with contamination indicator and cross validation scores of MCR-LLM with different number of components

Component	Correlation	CV similarity	
		Mean	Std
2	0.66	0.88	0.04
3	0.60	0.84	0.04
4	0.68	0.80	0.04
5	0.75	0.90	0.03
6	0.49	0.85	0.05
7	0.44	0.77	0.04
8	0.54	0.78	0.10

Looking at the loading profiles, the cross-validation, the correlation and the effective rank of 5.8, five components were finally chosen for analysis. With fewer than five components, MCR-LLM was not able to discriminate between all the different major contamination patterns, thus grouping them together. When randomly removing blocks of data during cross-validation, groups were extracted differently depending of which time periods were removed. This explains the lower similarity scores obtained. On the other hand, when too many components were used, the algorithm tried to separate the contamination patterns into too many different patterns. The lower similarity scores obtained with more than five components supports this observation. When randomly removing blocks of data, the MCR-LLM decomposition with more than five components did not extract the same loadings because the subtler patterns were randomly removed. Using more than five components therefore lead to less robust solutions.

The loading and contribution profiles from the MCR-LLM decomposition with five components are shown in Figure 4.12. For visualization purposes, contribution time profiles were averaged weekly and are only shown for a six-month period (Figure 4.12b). Additionally, to facilitate interpretation of the results, the spatial distribution of the most important extracted contamination components for each manufacturing area was integrated to a schematic of the site (Figure 4.12c). For every location, loading values of microbial recoveries variables that were higher than the average were represented by colored circles with the most important at the center and with sizes proportional to the intensity of the loading values. Alongside this information, inert particle count variables (IP) were represented by colored triangles in order to differentiate between variable types. With additional variable types, different visualization choices could be made.

The first MCR-LLM component was identified as the normal behavior component due to the high correlation value of 0.75 between its contribution profile and the complement of the sum of all microbial recoveries. The loading profile (Figure 4.12a) exhibited its highest values for variables associated to inert particle counts (IP), showing that high particle levels are not true indicators of a microbial contamination trend. This means that even when high particle levels are measured, there is no guarantee that the manufacturing area is out of microbial control. On the other hand, the normal behavior profile exhibited very low values for rooms R12, R5, R11 and R13 suggesting that positives recoveries in these rooms are direct indicators of problematic conditions.

The second extracted loading profile (orange) showed stronger values for PM, R1, R2, R3 and R4 areas. The PM variable refers to swabs performed on personnel hands, arms and chest whereas R1, R2, R3 and R4 are entrance or exit rooms to the aseptic area as shown in the schematic of the manufacturing area with green and red unidirectional arrows (Figure 4.12c). The second component reflected mostly contamination patterns associated to personnel activity but also highlighted the fact positive recoveries for personnel monitoring usually leads to contamination in rooms used to exit or enter the aseptic area of the plant. It is interesting to note that gowning for personnel happens in room R6 and therefore positive recoveries in room R1 are associated to contaminants before putting the gowns. We can also see, in the schematic, that the orange component is the second strongest for the gowning room R6 meaning that a significant portion of positive recoveries for personnel monitoring can be associated to the gowning procedure.

The third component (green) exhibited strong loading values for the two aseptic filling lines R7 and R9 and the corridor R13 used to access filling line R9. Additionally, the third component was the second strongest for personnel monitoring (PM). This loading profile reflected mostly contamination trends associated to similar activities performed in filling lines. Also, when filling lines are operational, personnel activity is usually more important in those areas, meaning that contamination in one place can more easily spread via the personnel; this explains the strong loading value for corridor R13.

The fourth component (blue) was mostly associated to the upper aseptic area as shown with the spatial distribution of the blue loading on the schematic. Upper section rooms such as R11, R2, R10, R4 exhibited strong loadings values for both microbial levels and particle counts as demonstrated with the strong presence of blue circles and triangles. Positive microbial recoveries and high inert particle counts in one area of the upper aseptic area usually means problematic operating conditions for the surroundings rooms. This could be due to many factors such as similar activities and cleaning practices in that area of the aseptic area. Additionally, most of the exit rooms (R2, R3, R4) also showed relatively strong expression of the fourth component for both microbial and particle counts. The MCR-LLM data analysis allowed for identification of this contamination pattern and a more in-depth investigation would be necessary to correctly identify the root cause.

The fifth extracted component (red) exhibited its strongest loading values for areas R5, R6, R12 and R13 located in the lower section of the aseptic core as shown in the schematic. As with the fourth component, the fifth component also showed strong values for inert particle counts in that same area. Spatial interpretation of this component is very similar to the previous component but for a different section of the manufacturing area. Positive microbial recoveries and high particle counts in one room of the lower aseptic core usually means problematic operating conditions for the surroundings rooms. It is interesting to note that particle counts in both filling lines were associated to the fifth component (red triangles) instead of the green microbial component for the microbial recovery variables. This is probably due to the air handling system that is shared for all the lower section of the aseptic area as well as the air exchanges happening during openings of the doors to get to both filling lines.

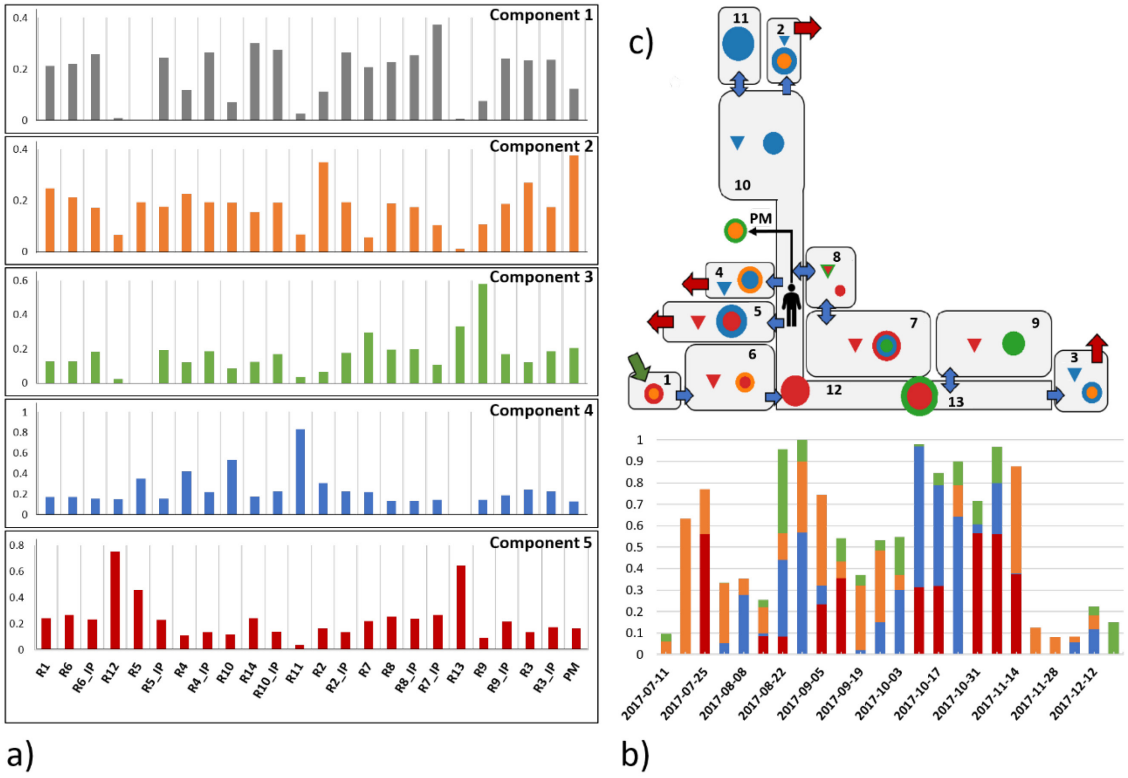

Figure 4.12 : Loading profiles (a), spatial distribution of important components for each area (b) and contribution time profiles (c) for the MCR-LLM data decomposition with the EM dataset

Based on the information of the loading matrix, contribution time profiles were used to identify when each contamination pattern was at its strongest (Figure 12b). For each weekly period, the magnitude of the stacked colored bars showed if the environmental monitoring dataset exhibited higher trends and gives additional insight on the type of contamination pattern. For example, for the three consecutive weeks ranging from 2017-10-31 to 2017-11-14 it is easy to describe the microbial level seen in each area and to associate such levels to a certain number of components, thus enabling a better understanding of the microbial level. As such, although the environment remained under control throughout the study period, an upward trend seen in some rooms identified by this type of analysis could enable appropriate action to be taken before the microbial levels are outside of the limits.

Overall, MCR-LLM data decomposition produced easily interpretable results that highlighted the presence of spatial patterns in the environmental monitoring dataset. The loading profiles extracted different types of problematic periods associated to different

variables and locations while the contribution time profiles greatly facilitated data visualization and interpretation.

4.6. Conclusions

This work described the use of MCR-ALS and MCR-LLM data decomposition techniques for non-spectral datasets. Many industries are generating very large amounts spatiotemporal data due to increased monitoring that can be studied to improve process understanding. While most datasets can be analyzed with standard data reduction algorithms, environmental monitoring datasets from pharmaceutical sites usually pose major challenges because of low signal to noise ratio.

The advantages of using MCR-LLM over MCR-ALS data reduction were demonstrated with three different noisy datasets. The LLM variant was shown to perform meaningful data reduction and greatly facilitate data interpretation while MCR-ALS struggled to produce interpretable results.

The first dataset consisted of a set of 1797 handwritten digits written onto 8×8 grids synthetically corrupted with variable levels of noise to assess its impact on data decomposition. MCR-LLM was shown to quantitatively outperform MCR-ALS for all levels of noise while maintaining meaningful results even with very noisy data.

The second dataset consisted in meteorological data from 23 different American weather stations over a two-month period during the hurricane season. The hurricane dataset highlighted the shortcomings of MCR-ALS in dealing with noisy spatiotemporal datasets with poor data understanding and result interpretability. On the other hand, MCR-LLM produced meaningful loading and contribution profiles that extracted typical types of weather conditions and allowed for a thorough interpretation of the result.

The third dataset used in this work came from an environmental monitoring program of a pharmaceutical manufacturing site with daily microbial samples in various locations for a period of three years. The MCR-LLM data reduction produced meaningful and easily interpretable results. The analysis outlined clear spatial patterns that could be visualized with a representation of the loading matrix on a schematic of the manufacturing floor.

Additionally, the contribution time profiles greatly facilitated visualization of temporal contamination dynamics.

This paper demonstrated the usefulness of using MCR-LLM for the analysis of challenging and noisy spatiotemporal datasets. For the pharmaceutical industry, this tool could be used to significantly improve environmental monitoring data understanding, and additional variables and parameters could also be added to broaden the scope of the analysis.

Acknowledgments:

This project was financed by MITACS Accelerate (application number IT12287) and the Pfizer Industrial Chair on PAT (Process Analytical Technologies). I would also like to thank Angela Moore, Alexander Maben and George Moore from Pfizer Global Supply for their help in collecting data.

5. CONCLUSION

5.1. Sommaire

Ce projet de recherche a abordé la problématique des données spatio-temporelles dans le cadre d'un processus industriel complexe. Les caractéristiques générales de ce type de problème sont rappelées ci-dessous :

- Processus complexe qui dépend de nombreux paramètres
- Données spatio-temporelles avec une résolution (spatiale et temporelle) variable
- Relations et dépendances complexes entre les variables
- Influence de facteurs difficilement mesurables qui peuvent impacter la modélisation
- Présence de bruit élevé dans les données
- Événements peu fréquents (rares) en raison de la nature des données

Dans un premier temps, l'étude des données environnementales à l'aide d'outils adaptés aux séries temporelles a permis de mettre en évidence des patrons et tendances de contamination au sein des données. Une approche couramment utilisée pour l'analyse de données temporelles est l'évaluation des similarités à l'aide d'indices tels que la distance euclidienne, la corrélation ou le dynamic time warping. En combinant la corrélation de Pearson avec le dynamic time warping, un indice de similarité représentatif des dynamiques dans les données a été obtenu. Une matrice de corrélation mettant en évidence les relations entre les différentes variables a été calculée et utilisée afin de construire un graphique à nœud permettant de facilement visualiser les dynamiques spatiales de contamination des zones de productions.

Les travaux de recherche ont ensuite permis de démontrer l'efficacité de l'analyse multivariée et plus particulièrement de l'algorithme MCR-LLM à analyser des données bruitées avec des dynamiques spatio-temporelles. Cette nouvelle approche a fait ressortir des patrons et tendances de contamination clairs. L'utilisation de cet outil à fortement faciliter l'interprétation des données de contrôle environnemental en mettant en évidence les principales dynamiques spatiales présente. L'algorithme peut aussi être utilisé sur d'autres jeux de données avec des caractéristiques spatio-temporelles similaires. Une visualisation temporelle simplifiée des données a aussi été produite avec la décomposition

MCR. Cet outil peut être utilisé avec de nouvelles données afin de comparer les dynamiques récentes avec les patrons et tendances des données historiques.

Pour le partenaire industriel, le projet de recherche a fourni un outil permettant l'analyse des données de contrôle environnemental. L'utilisation de cet outil permet d'améliorer la compréhension des données et facilite grandement l'interprétation des résultats avec des visualisations simples et concises. Les bénéfices pour le partenaire industriel sont résumés et catégorisés dans le tableau suivant. Trois types de bénéfices ont été identifiés en fonction de l'impact de ceux-ci.

Tableau 5.1: Bénéfices du projet pour le partenaire

Technologiques	Économiques	Sociaux
Simplification de l'analyse des données de contrôle environnemental	Optimisation du temps d'analyse et de manipulation des données	Réduction du risque de contamination des produits pharmaceutiques
Compréhension approfondie du processus de contrôle environnemental	Réductions des coûts en lien avec une diminution des périodes de déviations environnementales	
	Réduction des rejets	

5.2. Contributions originales

Ce projet de recherche contribue à la science en présentant de nouvelles approches pour l'analyse et la compréhension de jeux de données spatio-temporelles.

Dans un premier temps, le document présente une méthodologie plus spécifique aux données de contrôle environnemental en milieu industriel pharmaceutique. Les données sont analysées à l'aide d'un nouvel indice de similarité, développé dans le cadre du projet, qui utilise une combinaison de la corrélation de Pearson et le dynamic time warping.

Dans un deuxième temps, le projet de recherche contribue en présentant une nouvelle approche multivariée pour l'étude de données spatio-temporelle fortement bruitée à l'aide de l'algorithme MCR-LLM. Cette contribution sous forme d'un article scientifique apporte

une solution au problème de l'analyse de données spatio-temporelles (de façon générale) à faible rapport signal sur bruit.

Le projet de recherche a finalement permis de développer des outils de visualisation pour faciliter l'étude et l'interprétation des analyses sur les données de contrôle environnemental pour le partenaire industriel. Ces outils seront utiles pour la compréhension en profondeur des données, le suivi en continu et la réplication éventuelle des analyses d'autres sites.

5.3. Perspectives

Les travaux de recherche présentés dans ce mémoire pourraient bénéficier d'un approfondissement au niveau des variables incluses dans l'analyse des données de contrôle environnemental. En effet, l'ajout d'informations sur les nettoyages, les déplacements du personnel ou le type de nettoyant utilisé permettrait d'élargir la portée des analyses et des conclusions.

De façon générale, il serait aussi intéressant d'évaluer la possibilité d'ajouter un aspect prédictif à l'analyse MCR-LLM pour des données spatio-temporelles. Dans le cas des données de contrôle environnemental, le potentiel prédictif a été évalué comme faible en raison de la nature imprévisible des comportements humains et des événements sporadiques de contamination. Cependant, pour d'autres jeux de données spatio-temporelles, cet aspect pourrait apporter beaucoup de valeur.

A. ANNEXES

A.1 Grades pharmaceutiques et types d'analyses environnementales

Les instances gouvernementales à travers le monde utilisent 2 systèmes de cotation différents (mais équivalent) afin de définir le niveau de contrôle requis dans les différentes zones de production pharmaceutique. La FDA utilise une terminologie qui se base sur des classes ISO, allant de 4.8 à 8 pour l'industrie pharmaceutique [52]. L'union européenne et plusieurs pays asiatiques utilisent quant à eux un système de grades allant de A à D, avec le grade A étant le plus critique [4]. Le tableau ci-dessous met en évidence cette correspondance ainsi que le type d'opérations généralement effectuées dans les différents grades pharmaceutiques.

Grade pharmaceutique	Type d'opérations
Grade A / ISO 4.8	Le grade A défini les zones les plus à risque dans lesquelles le produit pharmaceutique est généralement directement exposé à l'air ambiant. Dans ces zones, un flot laminaire de l'air est requis afin d'éviter la propagation de contaminant.
Grade B / ISO 7	Le grade B définit les zones de support au Grade A. Ces zones se retrouvent généralement aux alentours des zones de grade A pour effectuer les opérations de support comme l'introduction d'équipements ou la manipulation de ceux-ci.
Grade C / ISO 8	Le grade C définit les zones moins critiques au processus aseptique mais qui nécessitent quand même un contrôle environnemental important. Les opérations comme la pesé des matières premières (avant les étapes de filtration) s'effectuent dans ces zones.
Grade D / ISO 8	Le grade D définit les zones les moins critiques pour le contrôle environnemental. Le nettoyages et l'entretien des équipements s'effectuent dans ces zones.

En fonction du grade d'une pièce ainsi que du type d'analyse effectué, différentes limites de détection (déclenchant une investigation) sont suggérées par les instances gouvernementales. Le tableau ci-dessous résume l'information relié aux limites de détection en fonction du type d'analyse et des grades [4].

Grade	Organisme dans l'air, actif (cfu/ m^3)	Organisme dans l'air, passif (cfu/plaque)	Organisme sur surface (cfu/25cm^2)	Organisme sur le personnel (cfu/main)	Particules inertes/m^3 ($\geq$ 5 µm)	Particules inertes/m^3 ($\geq$ 0.5 µm)
A	1 cfu				20	3520
B	10	5	5	5	2900	352000
C	100	50	25	NA	29000	3520000
D	200	100	50	NA	29000	3520000

Les analyses actives de microorganismes dans l'air s'effectuent avec un équipement d'aspiration d'air (SAS air sampler) qui fait passer l'air ambiant sur un plaque de type RODAC avec un milieu de culture permettant la prolifération de colonies bactériennes. Une fois la plaque exposée à un volume d'air précis, celle-ci est incubée dans un laboratoire pour une durée variant de 7 à 14 jours. Une fois la période d'incubation terminée, la plaque est analysée dans le but de compter le nombre de colonie (CFU) s'étant formé sur son milieu de culture.

Les analyses passives de microorganismes dans l'air utilisent aussi des plaques de culture de type RODAC. À la différence de la méthode active, les plaque sont directement exposées à l'air ambiant pour une durée prédéterminée pour ensuite être incubé et analyser de la même façon que les plaques issues de la méthode active. Ces plaques sont souvent utilisées pour échantillonner l'air dans les zones de grade A.

Les analyses sur les surfaces et sur le personnel utilisent aussi des plaques de culture de type RODAC en les déposant directement à plat (contact entre le milieu de culture et la surface) sur les surfaces à échantillonner.

Les analyses de particules inertes dans l'air utilisent un équipement d'aspiration d'air (Climet) et compte directement le nombre de particule passant devant une source de lumière.

Les fréquences d'échantillonnages doivent être déterminées par le site en fonction d'études de qualification et d'une revue périodique des données générées par le programme de contrôle environnemental [4]. De façon générale, plus le niveau de risque d'une pièce est élevé (grade A ou B par exemple), plus sa fréquence d'échantillonnage sera grande. Pour des raisons de confidentialité, les fréquences d'échantillonnage exactes utilisées par le partenaire industriel pour les différents grades pharmaceutiques ne peuvent être partagées. Cependant, de façon générale, l'ordre de grandeur est similaire d'une usine à l'autre avec une fréquence au moins journalière pour les grades A et B et une fréquence hebdomadaire pour les grades C et D.

A.2 Liste complète des valeurs des tests de stationnarité

Variables	p-value
PM	0.01684
R1	0.000296
R2	2.67E-05
R10_P	9.29E-06
R3	8.73E-06
R8	5.91E-06
R5_P	4.25E-06
R8_P	2.46E-06
R5	1.22E-06
R9	1.18E-06
R7_P	1.08E-06
R4	1.07E-06
R14_P	7.98E-07
R6	5.46E-07
R4_P	5.08E-07
R13	3.73E-07
R11	3.40E-07
R12	4.54E-08
R6_P	4.26E-08
R10	1.02E-08
R9_P	6.53E-09
R2_P	2.13E-09
R7	3.93E-10
R3_P	3.07E-11

A.3 Matrice de corrélation DTW

	R7	R3	R2	R8	R13	R10	R1	R12	R11	PM	R4	R9	R6	R5	R7_P	R3_P	R2_P	R8_P	R10_P	R4_P	R9_P	R6_P	R14_P	R5_P
R7	1.00	0.21	0.00	0.17	0.09	0.08	0.14	0.15	0.15	0.02	0.02	0.16	0.01	0.06	-0.01	0.09	0.11	0.06	0.05	0.18	-0.01	0.10	0.05	0.12
R3	0.21	1.00	0.31	0.12	-0.07	0.32	0.30	0.09	0.16	0.34	0.25	0.07	0.27	0.09	0.04	0.39	0.37	0.02	0.27	0.38	0.06	0.07	0.27	0.08
R2	0.00	0.31	1.00	0.20	0.06	0.32	0.22	0.08	0.22	0.20	0.29	0.02	0.29	0.19	0.08	0.17	0.21	0.01	0.22	0.24	0.18	0.18	0.14	0.06
R8	0.17	0.12	0.20	1.00	0.15	0.20	0.24	0.14	0.12	0.23	0.14	0.21	0.23	0.13	0.08	0.12	0.14	0.28	0.25	0.12	0.13	0.25	0.20	0.17
R13	0.09	-0.07	0.06	0.15	1.00	-0.02	0.13	0.33	-0.07	0.06	0.05	0.17	0.11	0.05	0.03	-0.02	-0.10	0.11	-0.08	-0.10	0.14	0.04	0.11	0.11
R10	0.08	0.32	0.32	0.20	-0.02	1.00	0.25	0.18	0.33	0.24	0.50	0.19	0.20	0.17	-0.02	0.40	0.33	-0.03	0.31	0.24	0.15	-0.05	0.10	-0.02
R1	0.14	0.30	0.22	0.24	0.13	0.25	1.00	0.22	0.16	0.38	0.26	0.18	0.40	0.11	0.08	0.18	0.24	0.15	0.22	0.22	0.22	0.20	0.22	0.32
R12	0.15	0.09	0.08	0.14	0.33	0.18	0.22	1.00	0.12	0.16	0.26	0.06	0.25	0.13	0.17	0.10	-0.13	0.13	0.06	-0.04	0.13	0.17	0.13	0.17
R11	0.15	0.16	0.22	0.12	-0.07	0.33	0.16	0.12	1.00	0.13	0.31	0.12	0.21	0.19	0.10	0.22	0.20	-0.08	0.10	0.24	0.07	0.02	0.17	0.10
PM	0.02	0.34	0.20	0.23	0.06	0.24	0.38	0.16	0.13	1.00	0.20	0.38	0.33	0.12	-0.06	0.16	0.16	0.17	0.20	0.24	0.35	0.13	0.06	0.24
R4	0.02	0.25	0.29	0.14	0.05	0.50	0.26	0.26	0.31	0.20	1.00	0.12	0.25	0.07	0.03	0.33	0.25	0.03	0.30	0.26	0.15	0.04	0.10	0.19
R9	0.16	0.07	0.02	0.21	0.17	0.19	0.18	0.06	0.12	0.38	0.12	1.00	0.14	0.00	-0.04	0.23	0.13	0.15	0.16	0.19	0.10	0.13	0.06	0.21
R6	0.01	0.27	0.29	0.23	0.11	0.20	0.40	0.25	0.21	0.33	0.25	0.14	1.00	0.18	0.30	0.14	0.05	0.32	0.18	0.10	0.11	0.46	0.28	0.44
R5	0.06	0.09	0.19	0.13	0.05	0.17	0.11	0.13	0.19	0.12	0.07	0.00	0.18	1.00	0.16	0.07	0.09	-0.03	0.00	0.07	0.08	0.14	-0.03	0.09
R7_P	-0.01	0.04	0.08	0.08	0.03	-0.02	0.08	0.17	0.10	-0.06	0.03	-0.04	0.30	0.16	1.00	-0.12	-0.16	0.29	0.13	-0.04	-0.01	0.30	0.14	0.13
R3_P	0.09	0.39	0.17	0.12	-0.02	0.40	0.18	0.10	0.22	0.16	0.33	0.23	0.14	0.07	-0.12	1.00	0.48	-0.08	0.32	0.45	0.16	-0.07	0.13	0.12
R2_P	0.11	0.37	0.21	0.14	-0.10	0.33	0.24	-0.13	0.20	0.16	0.25	0.13	0.05	0.09	-0.16	0.48	1.00	-0.11	0.39	0.82	0.27	0.08	0.09	0.16
R8_P	0.06	0.02	0.01	0.28	0.11	-0.03	0.15	0.13	-0.08	0.17	0.03	0.15	0.32	-0.03	0.29	-0.08	-0.11	1.00	0.11	-0.08	-0.14	0.27	0.12	0.30
R10_P	0.05	0.27	0.22	0.25	-0.08	0.31	0.22	0.06	0.10	0.20	0.30	0.16	0.18	0.00	0.13	0.32	0.39	0.11	1.00	0.41	0.10	0.14	0.11	0.19
R4_P	0.18	0.38	0.24	0.12	-0.10	0.24	0.22	-0.04	0.24	0.24	0.26	0.19	0.10	0.07	-0.04	0.45	0.82	-0.08	0.41	1.00	0.21	0.10	0.10	0.22
R9_P	-0.01	0.06	0.18	0.13	0.14	0.15	0.22	0.13	0.07	0.35	0.15	0.10	0.11	0.08	-0.01	0.16	0.27	-0.14	0.10	0.21	1.00	0.13	0.03	0.23
R6_P	0.10	0.07	0.18	0.25	0.04	-0.05	0.20	0.17	0.02	0.13	0.04	0.13	0.46	0.14	0.30	-0.07	0.08	0.27	0.14	0.10	0.13	1.00	0.08	0.61
R14_P	0.05	0.27	0.14	0.20	0.11	0.10	0.22	0.13	0.17	0.06	0.10	0.06	0.28	-0.03	0.14	0.13	0.09	0.12	0.11	0.10	0.03	0.08	1.00	0.05
R5_P	0.12	0.08	0.06	0.17	0.11	-0.02	0.32	0.17	0.10	0.24	0.19	0.21	0.44	0.09	0.13	0.12	0.16	0.30	0.19	0.22	0.23	0.61	0.05	1.00

A.4 Code Python

```
# -*- coding: utf-8 -*-
"""
Created on Wed Sep 25 09:49:24 2019

@author: VIELFA
"""

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import tkinter as tk
from tkinter import filedialog
import re
from dateutil.parser import parse
from fct_IMPORT_CSV import import_csv

def load_past():

    raw_data = import_csv()

    data = raw_data.copy()

    data.rename(columns={'SAMPLED_DATE' : 'Sampled Date','ANALYSIS' : 'Analysis', 'NAME': 'Component',
                         'BUILDING' : 'Building','ROOM': 'Room','VALUE':'Result' },inplace=True)

    data['Sampled Date'] = pd.to_datetime(data['Sampled Date'],format="%m/%d/%Y %H:%M")
    data['Sampled Date'] = data['Sampled Date'].dt.strftime('%Y-%m-%d')
    data = data.set_index('Sampled Date').sort_index()

    keep_col =
['Analysis','Component','Result','Building','Room','SPECIFICATION_ID','SmpDESCRIPTION','UNITS','ID_NUMERIC','TstSTATUS']

    data = data[keep_col]

    Y = data.copy()

    Y=Y[(~Y.Result.isnull())]

    #filtre NVAIR
    Y.loc[(Y.Component.str.contains("5.0")) & (Y.Analysis == 'NVAIR'), 'Analysis'] = "NVAIR5.0"
    Y.loc[(Y.Component.str.contains("0.5")) & (Y.Analysis == 'NVAIR'), 'Analysis'] = "NVAIR0.5"
    Y=Y[~Y.Component.str.contains("low")]

    Y = Y[~(Y.SmpDESCRIPTION.str.contains("Investigational",na=False))]
    Y = Y[~(Y.SmpDESCRIPTION.str.contains("OAR-",na=False))]
    Y = Y[~(Y.SmpDESCRIPTION.str.contains("Requalification",na=False))]
    Y = Y[~(Y.SmpDESCRIPTION.str.contains("Testing",na=False))]
    Y = Y[~(Y.SmpDESCRIPTION.str.contains("testing",na=False))]
    Y = Y[~(Y.SmpDESCRIPTION.str.contains("TSR",na=False))]

    Y.loc[(Y.UNITS == "CFU/plate") & (Y.Analysis == "VAIR") , 'Analysis'] = "SETTLE"

    Y.loc[(Y.Analysis == 'RODAC') & (Y.SPECIFICATION_ID == 'EXCHEST'), 'Analysis'] = "CHEST"

    Y = Y[Y['Analysis'].str.contains("RODAC",na=False) | Y['Analysis'].str.contains("NVAIR",na=False) |
          Y['Analysis'].str.contains("FINGER",na=False) | Y['Analysis'].str.contains("UNIFORM",na=False) |
          Y['Analysis'].str.contains("CHEST",na=False) | Y['Analysis'].str.contains("VAIR",na=False)
          | Y['Analysis'].str.contains("SETTLE",na=False)]

    Y.Analysis = Y.Analysis.map(lambda x: re.sub('YM','_YM', x))

    #Room Building
    Y.Building = Y.Building.astype(str).map(lambda x: re.sub('B-','', x))
    Y.Room = Y.Room.astype(str).map(lambda x: re.sub('CORE','EXIT', x))

    Y.loc[Y.Room.str.contains('R2-') & (Y.Building =='nan'), 'Building'] = "R2"
```

```
Y.loc[(~(Y.Room.str.contains('R2-')) & (Y.Building =='nan')) , 'Building'] = "R1"

Y.loc[(Y.Building == 'R1') & (Y.Room != 'R1-EXIT'),'Room'] = 'R1-'+Y[(Y.Building == 'R1') & (Y.Room != 'R1-EXIT')].Room

Y = Y[(Y.Building == 'R1') | (Y.Building == 'R2')]

#YM or not
Y.loc[(Y.Analysis.str.contains('YM')),'Component'] = 'YM'
Y.loc[~(Y.Analysis.str.contains('YM')),'Component'] = 'Count'

#BQA
Y.loc[Y['SmpDESCRIPTION'].str.contains('BQA',na=False), 'Analysis'] += '_BQA'

return Y
```

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 14 15:01:13 2020

@author: VIELFA
"""

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import tkinter as tk
from tkinter import filedialog
import re
from dateutil.parser import parse
from scipy.signal import savgol_filter
from fct_Load_Past import load_past
import shelve
from fct_FILL_MATRIX import fill_matrix
from dtaidistance import dtw
import networkx as nx
import matplotlib

#%% Fucntions

def remove_columns(dataframe,th):
    percent = np.zeros(len(dataframe.T))
    it = 0;
    for k in dataframe.columns:
        percent[it] = (dataframe[k].isnull().sum() /  len(dataframe[k]))*100
        if percent[it] > th :
            dataframe = dataframe.drop([k], axis = 1)
        it +=1
    return dataframe

def remove_rows(dataframe,th):
    rmv = dataframe.T
    for i in rmv.columns:
        d = (rmv[i].isnull().sum() / len(rmv[i]))*100
        if d > th :
            rmv = rmv.drop([i], axis = 1)
    dataframe = rmv.T
    return dataframe

def group_data(data,tot,opt,buil):
    if buil == 2:
        B = 'R2'
    elif buil == 1:
        B = 'R1'

    avg = data.copy()
    avg = avg.swaplevel(0, 1, axis=1)
    R2 = avg[B]

    if opt == 1:
        #Rooms QUANT
        R2 = avg[B]
        median = R2.median()
        std = R2.std()
        for j in range(len(R2.columns)):
            for k in range(len(R2)):
                if R2.iloc[k][j] > (median[j]+(3*std[j])):
                    R2.iloc[k][j] = median[j]+(3*std[j])
        R2 = np.log1p(R2)
        R2 = (R2 - R2.min())/R2.std()
        group_rooms = R2
```

```
        group_rooms.columns = [ '_'.join(x) for x in group_rooms.columns]

    if opt == 2:

        tot = tot.swaplevel(0, 2, axis=1)
        tot = tot.drop(['NVAIR5.0','NVAIR0.5'],axis=1)
        tot = tot.swaplevel(0, 2, axis=1)
        tot = tot.swaplevel(0, 1, axis=1)
        tot_room = tot[B]

        #Rooms CRR
        R2 = avg[B]
        R2 = R2.swaplevel(0, 1, axis=1)
        R2 = R2.drop(['NVAIR5.0','NVAIR0.5'],axis=1)
        #R2['FINGER'] = R2['FINGER']/2
        R2 = R2.swaplevel(0, 1, axis=1)
        R2_swap =  R2.swaplevel(0, 1, axis=1)
        room_r2 = list(set(R2_swap.columns.droplevel()))
        unit_R2 = []

        for i in room_r2:
            unit = R2[i].sum(axis=1,min_count=1)/tot_room[i].sum(axis=1,min_count=1)
            unit_R2.append(unit)

        group_rooms = R2
        group_rooms.columns = [ '_CRR_'.join(x) for x in group_rooms.columns]

    if opt == 0:
        #Rooms AA
        R2 = avg[B]
        R2 = R2.swaplevel(0, 1, axis=1)
        R2 = R2.drop(['NVAIR5.0','NVAIR0.5'],axis=1)
        #R2['FINGER'] = R2['FINGER']/2
        R2 = R2.swaplevel(0, 1, axis=1)
        R2_swap =  R2.swaplevel(0, 1, axis=1)
        room_r2 = list(set(R2_swap.columns.droplevel()))

        group_rooms = R2
        group_rooms.columns = [ '_AA_'.join(x) for x in group_rooms.columns]

    return group_rooms

#%% Load data

#Y = load_past()

#%% Organize data

#list_result_map = ["MAX","MEAN","LENGHT","NON_ZERO"]
#list_Y_len, list_Y_max, list_Y_mean, list_Y_nozero, list_x, room, list_result = crosstab_data(Y)

#%% Save organized data in list_result_new.out file

#filename='C:/Users/VIELFA\Desktop/RM_Recent Data/list_result_new.out'
#my_shelf = shelve.open(filename,'n')
#my_shelf['list_result'] = list_result
#my_shelf.close()

#%% Load saved organized data

filename='C:/Users/VIELFA\Desktop/RM_Recent Data/list_result.out'
my_shelf = shelve.open(filename)
for key in my_shelf:
    globals()[key]=my_shelf[key]
my_shelf.close()

#%% Manipulations
```

```
list_result2 = list_result.copy()
for i in range(len(list_result2)):
    list_result2[i] = list_result2[i].loc['2015-07-01':'2018-07-01']
    list_result2[i] = remove_columns(list_result2[i],90)
    list_result2[i] = remove_rows(list_result2[i],90)

for i in list_result2:
    i['R2-158PM']['R2']['FINGER'] = pd.concat([i['R2-158PM']['R2']['FINGER'],i['R2-158']['R2']
['FINGER']],axis=1).sum(axis=1,min_count=1)
    i['R2-158PM']['R2']['FINGER_BQA'] = pd.concat([i['R2-158PM']['R2']['FINGER_BQA'],i['R2-158']['R2']
['FINGER_BQA']],axis=1).sum(axis=1,min_count=1)

#%% Nb of test performed for each test
names = list(set(list_result2[2].columns.get_level_values(0)))

list_tot =[]
for i in names :
    tot = list_result2[2][i]
    tot_room = pd.DataFrame(tot[tot.columns.drop(list(tot.filter(regex='NVAIR')) + list(tot.filter(regex='People'))+
list(tot.filter(regex='YM')))].sum(axis=1))
    tot_nvair = pd.DataFrame(tot[list(tot.filter(regex='NVAIR').columns)].sum(axis=1))

    tot_mix = pd.concat([tot_room,tot_nvair],axis=1)
    tot_mix.columns = [i,i+'_NVAIR']
    list_tot.append(tot_mix)

tot_test = pd.concat(list_tot,axis=1)
tot_test.columns = 'Total_'+tot_test.columns
tot_test.index = pd.to_datetime(tot_test.index)


#%% Group by building and rooms

#Select time frame
list_result_time = list_result2.copy()
for i in range(len(list_result2)):
    list_result_time[i] = list_result2[i].loc['2015-07-01':'2018-07-01']

Rooms_QUANT = group_data(list_result_time[1],list_result_time[2],1,2)
Rooms_CRR = group_data(list_result_time[3],list_result_time[2],2,2)

Nb_people = Rooms_QUANT.filter(regex='People')
YM_quant = Rooms_QUANT.filter(regex='YM')
YM_crr = Rooms_CRR.filter(regex='YM')
NVAIR_all = Rooms_QUANT.filter(regex='NVAIR')

CRR_var = Rooms_CRR.drop((list(Rooms_CRR.filter(regex='People').columns) + list(YM_crr.columns)),axis=1)
QUANT_var = Rooms_QUANT.drop((list(Nb_people.columns) + list(YM_quant.columns) + list(NVAIR_all.columns)),axis=1)


#%% Map of missing data

#missing_matrix = Rooms_QUANT.drop((list(Nb_people.columns) + list(YM_quant.columns)),axis=1).filter(regex='R2')
#miss_map = ~missing_matrix.isnull()*1
##miss_map = miss_map.astype(np.float)
#plt.imshow(~miss_map, cmap="gray", interpolation='nearest', aspect = "auto")
#plt.title('Missing data in white')
#ax = plt.gca();
#ax.set_xticks(np.arange(0, len(missing_matrix.columns), 1));
##ax.set_xticklabels(list(missing_matrix),rotation=90,fontsize=5)
#plt.show()

#%% Data Pre-processing

data_t = CRR_var.drop(CRR_var.filter(regex='R2-2').columns,axis=1)
data_t = remove_columns(data_t,50)
data_t.drop(data_t.filter(regex='R2-158_CRR_FI').columns,axis=1,inplace=True)
```

```
data_t.drop('2018-01-02',axis=0,inplace=True)

data_n = NVAIR_all.drop(NVAIR_all.filter(regex='R2-2').columns,axis=1)
data_n = remove_columns(data_n,50)

period = '7d'
min_per = 4
shift = -3

#CRR rooms
t_lit =[]
names_rooms = list(data_t.columns)
for i in range(len(data_t.columns)):
    names_rooms[i] = names_rooms[i].split('_', 1)[0]
names = list(set(names_rooms))
data_t.index = pd.to_datetime(data_t.index)

for i in names :
    crr_a = data_t.filter(regex=re.escape(i)+'_')
    dat = pd.DataFrame((crr_a.sum(axis=1,min_count=1)).rolling(period,min_per).sum()).shift(shift)
    dat[(dat > dat.mean()+3*dat.std())] =  dat*0 + dat.mean()+3*dat.std()
    dat.columns = [i]
    t_lit.append(dat)

crr_t = pd.concat(t_lit,axis=1)

crr_t['R2-EXIT'].add(crr_t['R2-158PM'],fill_value=0)
crr_t = crr_t.drop('R2-158PM',axis=1)

crr_t = crr_t
crr_t = crr_t/crr_t.std()

#NVAIR
n_lit =[]
names_rooms = list(data_n.columns)
for i in range(len(data_n.columns)):
    names_rooms[i] = names_rooms[i].split('_', 1)[0]
names = list(set(names_rooms))
data_n.index = pd.to_datetime(data_n.index)

for i in names :
    nvair = data_n.filter(regex=re.escape(i)+'_')
    d_nvair = pd.DataFrame(nvair.mean(axis=1).rolling(period,min_per).mean()).shift(shift)
    d_nvair[(abs(d_nvair-d_nvair.mean()) > 3*d_nvair.std())] =  d_nvair*0 + d_nvair.mean() +
np.sign(d_nvair-d_nvair.mean())*3*d_nvair.std()
    d_nvair.columns = [i+'_NVAIR']
    n_lit.append(d_nvair)

nvair_n = pd.concat(n_lit,axis=1)
nvair_n = nvair_n
nvair_n = (nvair_n - nvair_n.mean())/nvair_n.std()
nvair_n = nvair_n + abs(nvair_n.min())

# Weighing

Nb_tests = tot_test.loc['2015-07-01':'2018-07-01']
test = Nb_tests['Total_'+crr_t.columns].sum()

Nb_tests['Total_'+crr_t.columns].sum()
for i in crr_t.columns:
    crr_t[i] *= Nb_tests['Total_'+crr_t.columns].sum()['Total_'+i]**0.333

for i in nvair_n.columns:
    nvair_n[i] *= Nb_tests['Total_'+nvair_n.columns].sum()['Total_'+i]**0.333

w_data = pd.concat([crr_t,nvair_n],axis=1)
```

```
# Remove rooms that are not in the aspectic core
w_data =w_data.drop(w_data.filter(regex='174').columns,axis=1)
w_data =w_data.drop(w_data.filter(regex='159').columns,axis=1)
w_data =w_data.drop(w_data.filter(regex='181').columns,axis=1)
w_data =w_data.drop(w_data.filter(regex='182').columns,axis=1)

w_data = remove_rows(w_data,50)

# Fill missing values with PCA
dummy,w_data  = fill_matrix(w_data)
w_data[w_data<0] = 0

X = np.array(w_data)

#%% Dataviz

#plt.figure()
#Nb_tests['Total_'+'R2-EXIT'].plot()
#w_data['R2-157'].plot()
#w_data['R2-131'].plot()
#w_data['R2-138_NVAIR'].plot()

#data[0].plot()
#(crr_t['R2-131']/crr_t['R2-131'].max()).plot()
#(crr_t['R2-EXIT']/crr_t['R2-EXIT'].max()).plot()
#(nvair_n['R2-158_NVAIR']/nvair_n['R2-158_NVAIR'].max()).plot()
#(nvair_n['R2-130_NVAIR']/nvair_n['R2-130_NVAIR'].max()).plot()

#plt.figure()
#w_data['R2-158'].plot()
#w_data['R2-EXIT'].plot()

#data_t.filter(regex=re.escape('R2-158')+'_').plot()
#crr_t['R2-158'].plot(linewidth=3)
#plt.legend(['RODAC','SETTLE','VAIR','Indicateur global roulant'],loc='upper right')
#data_n.filter(regex=re.escape('R2-158')+'_').plot()

#%% Stationarity test

#from statsmodels.tsa.stattools import adfuller, kpss
#st_d = w_data['R2-EXIT']
#st_d = st_d.dropna()
#st_d.plot()
#plt.legend(['PM_all'])
#result = adfuller(st_d,autolag='AIC')
#print('ADF Statistic: %f' % result[0])
#print('p-value: %f' % result[1])
#print('Critical Values:')
#for key, value in result[4].items():
# print('\t%s: %.3f' % (key, value))
##Less equals stationary
#
##All values
#p_values = []
#for i in w_data.columns:
#    result = adfuller(w_data[i],autolag='AIC')
#    p_values.append(result[1])
#
#st_stat = pd.DataFrame(p_values)
#st_stat.index=w_data.columns


#%% Autocorrelation

#All values
#autocorr_values = []
#for i in w_data.columns:
```

```
#    max_autocorr = []
#    for k in [8,9,10,11,12,13,14]:
#        max_autocorr.append(w_data[i].autocorr(lag=k))
#    autocorr_values.append(max(max_autocorr))
#
#autocorr_stat = pd.DataFrame(autocorr_values)
#autocorr_stat.index=w_data.columns


#%% DTW correlation

#Problematic rooms
Data_look = w_data
Prob = pd.DataFrame([CRR_var[i][CRR_var[i] > 0].count() for i in CRR_var.columns])
Prob.index=[i.split('_', 1)[0] for i in CRR_var.columns]
Prob = Prob.groupby(Prob.index).sum()
Prob = Prob.loc[list(set(Prob.index).intersection(w_data.columns))]

#Tested rooms
Tested = tot_test.loc['2015-07-01':'2020-07-01']
nb_test = pd.DataFrame([Tested[i].sum() for i in 'Total_' + Data_look.columns])
nb_test.index= Data_look.columns
nb_test.drop(Data_look.filter(regex='NVAIR'),axis=0,inplace=True)

Recap = pd.concat([Prob,nb_test],axis=1)
Recap.columns = ['Nb_positive','Nb_test']

#Normal correlation matrix
corr_matrix = w_data.corr(method = 'pearson', min_periods = len(w_data)/2)


data_mat = w_data.copy()
corr_dtw = data_mat.corr(method = 'pearson', min_periods = len(w_data)/1.5)*np.nan
dtw_corr_pattern = []
timestamps = []
for i in corr_dtw.columns:
    for j in corr_dtw.index:

        s1 = np.array(data_mat[i].fillna(data_mat[i].mode()[0]))
        s2 = np.array(data_mat[j].fillna(data_mat[j].mode()[0]))

        path = dtw.warping_path(s1, s2,window=2)

        indx_1 = [i[0] for i in path]
        indx_2 = [i[1] for i in path]

        new_1 = data_mat[i].iloc[indx_1]
        new_2 = data_mat[j].iloc[indx_2]

        new_1 = new_1.reset_index()
        new_2 = new_2.reset_index()

        corr_dtw.loc[i,j] = new_1.iloc[:,1].corr(new_2.iloc[:,1])

        dtw_corr_pattern.append(new_1.iloc[:,1].rolling(90,45,center=True).corr(new_2.iloc[:,1]))
        timestamps.append(new_1.iloc[:,0])

#%% Top 10 correlation DTW pairs and bootstrapping

#top10_corr = corr_dtw.stack()
#
#v1 = 'R2-180A'
#v2 = 'R2-153_NVAIR'
#
#s1 = np.array(data_mat[v1].fillna(0))
#s2 = np.array(data_mat[v2].fillna(0))
#path = dtw.warping_path(s1, s2,window=2)
```

```
#
#indx_1 = [i[0] for i in path]
#indx_2 = [i[1] for i in path]
#new_1 = data_mat[v1].iloc[indx_1]
#new_2 = data_mat[v2].iloc[indx_2]
#new_1 = new_1.reset_index()
#new_2 = new_2.reset_index()
#
#x = np.array(new_1[v1]).reshape((len(new_1[v1]),))
#y = np.array(new_2[v2])
#
#n = len(x)
#a = 0.05 # alpha
#
#print('--------------')
#print('Bootstrap')
#nb = 1000
#boot = np.zeros((nb,X.shape[1]))
#boot = np.zeros(nb)
#for i in range(nb):
#    r = np.random.randint(0, n, n)
#    Xr = x[r]
#    Yr = y[r]
#
#    boot[i] = np.corrcoef(Xr,Yr)[0,1]
#
#boot.mean()
#
#ci = np.zeros((1,3))
#ci[0,0] = np.percentile(boot, a/2*100)
#ci[0,1] = np.percentile(boot, 50)
#ci[0,2] = np.percentile(boot, (1-a/2)*100)
#
#print(np.round(ci,4))

#%% DTW corr example

#from dtaidistance import dtw_visualisation as dtwvis
#data_mat = w_data.loc['2010-02-20':'2020-04-22']
##data_mat = w_data.loc['2016-05-10':'2016-07-01']
#s1 = np.array(data_mat['R2-EXIT'].fillna(0))
#s2 = np.array(data_mat['R2-158'].fillna(0))
#path = dtw.warping_path(s1, s2,window=2)
#dtwvis.plot_warping(s1, s2, path)
#
#indx_1 = [i[0] for i in path]
#indx_2 = [i[1] for i in path]
#new_1 = data_mat['R2-EXIT'].iloc[indx_1]
#new_2 = data_mat['R2-158'].iloc[indx_2]
#new_1 = new_1.reset_index()
#new_2 = new_2.reset_index()
#
#plt.figure()
#data_mat['R2-158'].plot()
#data_mat['R2-EXIT'].plot()
#plt.legend(['R2-158','R2-EXIT'],loc='upper right')
#data_mat['R2-158'].corr(data_mat['R2-EXIT'])
#
#plt.figure()
#new_2['R2-158'].plot()
#new_1['R2-EXIT'].plot()
#plt.legend(['R2-158','R2-EXIT'],loc='upper right')
#new_2['R2-158'].corr(new_1['R2-EXIT'])
#
#d, paths = dtw.warping_paths(s1, s2, window=2)
#best_path = dtw.best_path(paths)
#dtwvis.plot_warpingpaths(s1, s2, paths, best_path)
```

```
#%% Relations intermittentes

indx_1 = list(corr_dtw.columns).index('R2-EXIT')
indx_2 = list(corr_dtw.columns).index('R2-158')

test1 = timestamps[indx_2 * len(corr_dtw) + indx_1]

test = dtw_corr_pattern[indx_2 * len(corr_dtw) + indx_1]
test[(~np.isfinite(test)) & (~test.isnull())] = np.nan
test[(test>1.1) | (test<-1.1) | (abs(test)<0.00001)]= np.nan
test.index = test1
test.plot()
plt.axhline(0, color='black', linestyle = '--')
plt.legend(['Correlation pattern : R2-EXIT vs R2-158'])
plt.xlabel('Time')
plt.ylabel('DTW correlation')

#%% Correlation Matrix clustering

import matplotlib.colors as colors
class MidpointNormalize(colors.Normalize):
 """
 Normalise the colorbar so that diverging bars work there way either side from a prescribed midpoint value)

 e.g. im=ax1.imshow(array, norm=MidpointNormalize(midpoint=0.,vmin=-100, vmax=100))
 """
 def __init__(self, vmin=None, vmax=None, midpoint=None, clip=False):
  self.midpoint = midpoint
  colors.Normalize.__init__(self, vmin, vmax, clip)

 def __call__(self, value, clip=None):
  # I'm ignoring masked values and all kinds of edge cases to make a
  # simple example...
  x, y = [self.vmin, self.midpoint, self.vmax], [0, 0.5, 1]
  return np.ma.masked_array(np.interp(value, x, y), np.isnan(value))

def plot_corr(df,size=10):
    # Compute the correlation matrix for the received dataframe
    corr = df

    # Plot the correlation matrix
    fig, ax = plt.subplots(figsize=(size, size))
    cax = ax.matshow(corr, cmap='RdYlGn', norm=MidpointNormalize(midpoint=corr.mean().mean(),vmin=corr.min().min(),
vmax=corr.max().max()))
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90,size=10);
    plt.yticks(range(len(corr.columns)), corr.columns,size=10);

    # Add the colorbar legend
    cbar = fig.colorbar(cax,aspect=20, shrink=.8)

matrix = corr_dtw#_short
data_corr = w_data#.loc[period_start:period_end]

corr_matrix2 = matrix[matrix['R2-EXIT'].notnull()]
#corr_matrix2 = corr_matrix2.drop(list(corr_matrix2.T.filter(regex='NVAIR').columns))
#corr_matrix2 = corr_matrix2.loc[list(corr_matrix2.T.filter(regex='NVAIR').columns)]
corr_matrix2 = corr_matrix2[list(corr_matrix2.index)]


# Corr matrix clustering
import scipy.cluster.hierarchy as spc
from scipy.cluster.hierarchy import dendrogram
corr_matrix3 = corr_matrix2.dropna(axis=1)


top_all = corr_matrix3.columns
```

```
pdist = spc.distance.pdist(corr_matrix3)
linkage = spc.average(pdist)
plt.figure()
dn = dendrogram(linkage,labels=top_all,leaf_rotation = 0,leaf_font_size =14,orientation = 'right', color_threshold = 1.15)
idx = spc.fcluster(linkage, 0.3 * pdist.max(), 'distance')

corr_mat_excel = data_corr[top_all]
columns = [corr_mat_excel.columns.tolist()[i] for i in list((np.argsort(idx)))]
corr_mat_excel =corr_mat_excel[columns]#.reindex_axis(columns, axis=1)

corr_red = corr_mat_excel.corr(method = 'pearson', min_periods =len(w_data)/1.5)

data_mat = corr_mat_excel
corr_red = corr_red*np.nan
for i in corr_red.columns:
    for j in corr_red.index:

        s1 = np.array(data_mat[i].fillna(0))
        s2 = np.array(data_mat[j].fillna(0))

        path = dtw.warping_path(s1, s2,window=2)

        indx_1 = [i[0] for i in path]
        indx_2 = [i[1] for i in path]

        new_1 = data_mat[i].iloc[indx_1]
        new_2 = data_mat[j].iloc[indx_2]

        new_1 = new_1.reset_index(drop=True)
        new_2 = new_2.reset_index(drop=True)

        corr_red.loc[i,j] = new_1.corr(new_2)

corr_red[corr_red>0.5] = 0.5
#corr_red[corr_red<0.3] = 0.3
plot_corr(corr_red, size=8)


#%% Node graph

Recap_use = Recap.copy()

#Transform it in a links data frame (3 columns only):
links = corr_dtw.stack().reset_index()
links.columns = ['var1', 'var2','value']

#Weight corr
links['weight'] = links.value*np.nan
for i in range(len(links)):
    if not 'NVAIR' in links.iloc[i,0] and not 'NVAIR' in links.iloc[i,1]:
        links.iloc[i,3] = (Recap_use.loc[links.iloc[i,0],'Nb_positive'] + Recap_use.loc[links.iloc[i,1],'Nb_positive'])/2
    elif not 'NVAIR' in links.iloc[i,0]:
        links.iloc[i,3] = (Recap_use.loc[links.iloc[i,0],'Nb_positive'] + Recap_use['Nb_positive'].mean())/2
    elif not 'NVAIR' in links.iloc[i,1]:
        links.iloc[i,3] = (Recap_use['Nb_positive'].mean() + Recap_use.loc[links.iloc[i,1],'Nb_positive'])/2
    else:
        links.iloc[i,3] = Recap_use['Nb_positive'].mean()
links['value'] = links['value']*1#links['weight']

# Keep only correlation over a threshold and remove self correlation (cor(A,A)=1)
th = 0.90
nvair_th = links.loc[(links['var1'] != links['var2']) & (links['var1'].str.contains('NVAIR')) &
(links['var2'].str.contains('NVAIR'))].quantile([th]).iloc[0,0]
room_th = links.loc[(links['var1'] != links['var2']) & ((~links['var1'].str.contains('NVAIR')) |
(~links['var2'].str.contains('NVAIR')))].quantile([th]).iloc[0,0]
```

```
links_filtered_nvair = links.loc[ (links['value'] > nvair_th) & (links['var1'] != links['var2']) & (links['var1'].str.contains('NVAIR')) &
(links['var2'].str.contains('NVAIR'))]
links_filtered_room = links.loc[ (links['value'] > room_th) & (links['var1'] != links['var2']) & ((~links['var1'].str.contains('NVAIR')) |
(~links['var2'].str.contains('NVAIR')))]
links_filtered = pd.concat([links_filtered_room,links_filtered_nvair],axis=0)

# Build your graph
G=nx.from_pandas_edgelist(links_filtered,'var1','var2','value')

edges,weights = zip(*nx.get_edge_attributes(G,'value').items())
weights = tuple([((1+abs(x))**2) for x in weights])
weights = tuple(x/max(weights)*2 +1 for x in weights)

node = pd.concat(t_lit,axis=1)
node.columns = node.columns.str.replace("_CRR", "")
node = pd.DataFrame(node.sum()).T
node_size = np.zeros(len(G.node))
indx = 0
for i in G.node:
    if 'NVAIR' in i:
        node_size[indx] = np.nan
    else:
        node_size[indx] = node[i]
    indx = indx+1
indx = 0
for i in G.node:
    if 'NVAIR' in i:
        node_size[indx] = np.nanmean(node_size)
    indx = indx+1

pos_dic = dict(G.node)
test = []
for key, value in pos_dic.items():
    if not 'NVAIR' in key:
        #pos_dic[key] = room_loc[key]
        test.append(Recap.loc[key,'Nb_positive'])
    else:
        #pos_dic[key] = room_loc[key]
        test.append(Recap['Nb_positive'].mean())

test = np.array(test)
test = ((test-test.min())/test.max())*300 + 50

#plt.figure()
#nx.draw(G,with_labels=True,node_color='orange', width=weights,edge_vmin = min(weights),
edge_vmax=max(weights),node_size=test, edge_color='grey', font_size=11)

# Slider node graph
from matplotlib.widgets import Slider

fig = plt.figure(figsize=(7, 7))
ax = fig.add_subplot(111)
fig.subplots_adjust(bottom=0.25)
nx.draw(G,with_labels=True,node_color='orange', width=weights,edge_vmin = min(weights),
edge_vmax=max(weights),node_size=test, edge_color='grey',font_size=11)

axcolor = 'lightgoldenrodyellow'
axtresh  = fig.add_axes([0.25, 0.15, 0.65, 0.03], facecolor=axcolor)
sth = Slider(axtresh, 'Similarity treshold', 0, 1, valinit=th,valfmt='%1.2f')

def update2(val):

    th = sth.val
    nvair_th = links.loc[(links['var1'] != links['var2']) & (links['var1'].str.contains('NVAIR')) &
(links['var2'].str.contains('NVAIR'))].quantile([th]).iloc[0,0]
```

```
    room_th = links.loc[(links['var1'] != links['var2']) & ((~links['var1'].str.contains('NVAIR')) |
(~links['var2'].str.contains('NVAIR')))].quantile([th]).iloc[0,0]

    links_filtered_nvair = links.loc[ (links['value'] > nvair_th) & (links['var1'] != links['var2']) & (links['var1'].str.contains('NVAIR')) &
(links['var2'].str.contains('NVAIR'))]
    links_filtered_room = links.loc[ (links['value'] > room_th) & (links['var1'] != links['var2']) & ((~links['var1'].str.contains('NVAIR'))
| (~links['var2'].str.contains('NVAIR')))]
    links_filtered = pd.concat([links_filtered_room,links_filtered_nvair],axis=0)

    # Build your graph
    G=nx.from_pandas_edgelist(links_filtered,'var1','var2','value')

    edges,weights = zip(*nx.get_edge_attributes(G,'value').items())
    weights = tuple([(1+abs(x))**2) for x in weights])
    weights = tuple(x/max(weights)*2 +1 for x in weights)

    node = pd.concat(t_lit,axis=1)
    node.columns = node.columns.str.replace("_CRR", "")
    node = pd.DataFrame(node.sum()).T
    node_size = np.zeros(len(G.node))
    indx = 0
    for i in G.node:
        if 'NVAIR' in i:
            node_size[indx] = np.nan
        else:
            node_size[indx] = node[i]
        indx = indx+1
    indx = 0
    for i in G.node:
        if 'NVAIR' in i:
            node_size[indx] = np.nanmean(node_size)
        indx = indx+1

    pos_dic = dict(G.node)
    test = []
    for key, value in pos_dic.items():
        if not 'NVAIR' in key:
            #pos_dic[key] = room_loc[key]
            test.append(Recap.loc[key,'Nb_positive'])
        else:
            #pos_dic[key] = room_loc[key]
            test.append(Recap['Nb_positive'].mean())

    test = np.array(test)
    test = ((test-test.min())/test.max())*300 + 50

    ax.clear()
    nx.draw(G,alpha = 0.9,with_labels=True,node_color='orange', width=weights,edge_vmin = min(weights),
edge_vmax=max(weights),node_size=test, edge_color='grey', font_size=10,ax=ax)

sth.on_changed(update2)
plt.show()


#%% MCR model

from mcrllm_new import mcrllm
decomposition = mcrllm(X,5,'phi')
decomposition.iterate(20)

# Get results
allS = decomposition.allS
S_final = decomposition.S
allC = decomposition.allC
C_final = decomposition.C
Sini = decomposition.Sini
```

```
allphi = decomposition.allphi

#%% MCR model viz
compo = pd.DataFrame(S_final.T)
compo.index = w_data.columns
var_name =list(w_data.columns)
var_name.sort()

compo =  compo.T[var_name]
test = compo.T
test = test.divide(test.sum(axis=1),axis=0)
test.plot.bar()

data = pd.DataFrame(C_final)
data.index= pd.to_datetime(w_data.index)
prob_comp = np.argmin(data.sum(axis=0))
data.plot()

test = data.resample('7d').mean()
#test.loc[test[0] > 0.5, [1,2,3,4]] = 0
test[0]=np.nan
import matplotlib.ticker as mticker
ax = test.plot.bar(stacked=True)
skip = len(test)//24
ticklabels = ['']*len(test)
ticklabels[::skip] = test.index[::skip].strftime('%Y-%m-%d')
ax.xaxis.set_major_formatter(mticker.FixedFormatter(ticklabels))

#%% MCR viz map

cp = 2
blobi = compo.T.copy()

blobi = blobi.divide(blobi.sum(axis=1),axis=0)
blobi[blobi<0.2] = 0
blob =  blobi
val=200

room_loc = {'R2-128' : (296,27),
 'R2-129' : (320,35),
 'R2-129_NVAIR' : (315,50),
 'R2-130' : (298,65),
 'R2-133' : (275,102),
 'R2-138' : (257,102),
 'R2-138_NVAIR' : (257,86),
 'R2-130A' : (321,101),
 'R2-130A_NVAIR' : (321,101),
 'R2-130_NVAIR' : (315,83),
 'R2-131' : (302,119),
 'R2-131_NVAIR' : (302,119),
 'R2-133_NVAIR' : (275,80),
 'R2-135' :(261,29) ,
 'R2-139' : (184,103),
 'R2-139A_NVAIR' : (133,85),
 'R2-139_NVAIR' : (184,80),
 'R2-151' : (53,85),
 'R2-153' : (85,120),
 'R2-151_NVAIR' : (53,102),
 'R2-153_NVAIR' : (65,120),
 'R2-156' : (288,154),
 'R2-156A' : (266,135),
 'R2-156A_NVAIR' : (248,133),
 'R2-156_NVAIR' : (288,177),
 'R2-157' : (320,200),
 'R2-157_NVAIR' : (320,165),
 'R2-180A' : (320,261),
 'R2-180A_NVAIR' : (320,277),
```

```
 'R2-158' : (296,234),
 'R2-158PM' : (312,209),
 'R2-158_NVAIR' : (296,255),
 'R2-159C_NVAIR' : (218,247),
 'R2-174' : (290,377),
 'R2-174_NVAIR' : (290,377),
 'R2-181' : (296,298),
 'R2-181_NVAIR' : (296,298),
 'R2-182' : (222,308),
 'R2-182_NVAIR' : (222,308),
 'R2-183' : (320,255),
 'R2-183_NVAIR' : (320,255),
 'R2-EXIT' : (211,160)}

from PIL import Image
I = Image.open('C:/Users/VIELFA\Desktop/Capture.png')
#I = Image.open('C:/Users/VIELFA\Desktop/map_schema1.png')
I = I.resize((441,345), 1)
p = np.asarray(I).astype('float')
fig, ax = plt.subplots()
ax.imshow(I)

x1=[]
y1=[]
for i in blob.index:
    x1.append(room_loc[i][1])
    y1.append(room_loc[i][0])

blob_t = blob.copy()

blob_t[cp] = 0

blo_min = blob_t[blob_t.gt(0)].idxmin(axis=1)

blo_size1 = np.pi*(blob_t.sum(axis=1)**2)
blo_color1 = blo_min.copy()

ind = 0
for i in blo_min:
    if np.isfinite(i) :
        blob_t[int(i)][ind] = 0
        #blob_t.iloc[ind,int(i)] = 0
    ind+=1

blo_size2 = np.pi*(blob_t.sum(axis=1)**2)
blo_min = blob_t[blob_t.gt(0)].idxmin(axis=1)
blo_color2 = blo_min.copy()


ind = 0
for i in blo_min:
    if np.isfinite(i) :
        blob_t[int(i)][ind] = 0
        #blob_t.iloc[ind,int(i)] = 0
    ind+=1

blo_size3 = np.pi*(blob_t.sum(axis=1)**2)
blo_min = blob_t[blob_t.gt(0)].idxmin(axis=1)
blo_color3 = blo_min.copy()


ind = 0
for i in blo_min:
    if np.isfinite(i) :
        blob_t[int(i)][ind] = 0
        #blob_t.iloc[ind,int(i)] = 0
    ind+=1
```

```
blo_size4 = np.pi*(blob_t.sum(axis=1)**2)
blo_min = blob_t[blob_t.gt(0)].idxmin(axis=1)
blo_color4 = blo_min.copy()


ind = 0
for i in blo_min:
    if np.isfinite(i) :
        blob_t[int(i)][ind] = 0
        #blob_t.iloc[ind,int(i)] = 0
    ind+=1

blo_size5 = np.pi*(blob_t.sum(axis=1)**2)
blo_min = blob_t[blob_t.gt(0)].idxmin(axis=1)
blo_color5 = blo_min.copy()


for i in range(len(blob.T)):
    #if not i == cp:
    if not i in blo_color1.values:
        #blo_color1[blo_color1[blo_color1.isna()].index[0]] = i
        blo_color1.loc['R2-139A_NVAIR'] = i
    if not i in blo_color2.values:
        blo_color2[blo_color2[blo_color2.isna()].index[0]] = i
    if not i in blo_color3.values:
        blo_color3[blo_color3[blo_color3.isna()].index[0]] = i
    if not i in blo_color4.values:
        blo_color4[blo_color4[blo_color4.isna()].index[0]] = i

    if not i in blo_color5.values:
        blo_color5[blo_color5[blo_color5.isna()].index[0]] = i


colors = ['tab:red','tab:green','white','tab:orange','tab:blue']


blo_size1_s = blo_size1.copy()
blo_size1_s.loc[blo_size1.T.drop(blo_size1_s.T.filter(regex='NVAIR').index).index]=0
blo_size2_s = blo_size2.copy()
blo_size2_s.loc[blo_size2_s.T.drop(blo_size2_s.T.filter(regex='NVAIR').index).index]=0
blo_size3_S = blo_size3.copy()
blo_size3_S.loc[blo_size3_S.T.drop(blo_size3_S.T.filter(regex='NVAIR').index).index]=0
blo_size4_S = blo_size4.copy()
blo_size4_S.loc[blo_size4_S.T.drop(blo_size4_S.T.filter(regex='NVAIR').index).index]=0
blo_size5_S = blo_size5.copy()
blo_size5_S.loc[blo_size5_S.T.drop(blo_size5_S.T.filter(regex='NVAIR').index).index]=0

blo_size1.loc[blo_size1.T.filter(regex='NVAIR').index] = 0
blo_size2.loc[blo_size2.T.filter(regex='NVAIR').index] = 0
blo_size3.loc[blo_size3.T.filter(regex='NVAIR').index] = 0
blo_size4.loc[blo_size4.T.filter(regex='NVAIR').index] = 0
blo_size5.loc[blo_size5.T.filter(regex='NVAIR').index] = 0


ax.scatter(x1,y1,s=blo_size1 *val, alpha=1, c=blo_color1,cmap=matplotlib.colors.ListedColormap(colors))
ax.scatter(x1,y1,s=blo_size2 *val, alpha=1, c=blo_color2,cmap=matplotlib.colors.ListedColormap(colors))
ax.scatter(x1,y1,s=blo_size3 *val, alpha=1, c=blo_color3,cmap=matplotlib.colors.ListedColormap(colors))
ax.scatter(x1,y1,s=blo_size4 *val, alpha=1, c=blo_color4,cmap=matplotlib.colors.ListedColormap(colors))
ax.scatter(x1,y1,s=blo_size5 *val, alpha=1, c=blo_color5,cmap=matplotlib.colors.ListedColormap(colors))

ax.scatter(x1,y1,s=blo_size1_s *val, marker = '^',alpha=1, c=blo_color1,cmap=matplotlib.colors.ListedColormap(colors))
ax.scatter(x1,y1,s=blo_size2_s *val, marker = '^',alpha=1, c=blo_color2,cmap=matplotlib.colors.ListedColormap(colors))
ax.scatter(x1,y1,s=blo_size3_S *val, marker = '^',alpha=1, c=blo_color3,cmap=matplotlib.colors.ListedColormap(colors))
ax.scatter(x1,y1,s=blo_size4_S *val, marker = '^',alpha=1, c=blo_color4,cmap=matplotlib.colors.ListedColormap(colors))
ax.scatter(x1,y1,s=blo_size5_S *val, marker = '^',alpha=1, c=blo_color5,cmap=matplotlib.colors.ListedColormap(colors))
```

```
fig.patch.set_visible(False)
ax.axis('off')


#%% MCR for new data

def c_cal(n_data, S_load):

    nbc = S_load.shape[0]
    old_c = np.ones([n_data.shape[0],S_final.shape[0]])/S_final.shape[0]

    c_new = np.zeros((n_data.shape[0],S_final.shape[0]))

    # on calcule les concentrations optimales pour chaque pixel par maximum likelihood
    for pix in range(n_data.shape[0]):
        sraw = S_load*np.sum(n_data,axis=1)[pix]
        c_new[pix,:] = mcrllm.pyPLM_new(nbc,sraw,n_data[pix,:], old_c[pix,:])

    # avoid errors (this part should not be necessary)
    c_new[np.isnan(c_new)] = 1/S_final.shape[0]
    c_new[np.isinf(c_new)] = 1/S_final.shape[0]
    c_new[c_new<0] = 0
    c_sum1 = np.array([np.sum(c_new,axis=1)]).T
    c_new = c_new/c_sum1

    old_c = c_new.copy()

    return old_c

test = c_cal(X[-60:,:],S_final)
plt.plot(test)

#%% Effective rank

X_rank = X-X.mean()

A = X_rank.T@X_rank

[P,D,PT] = np.linalg.svd(A)
D = np.diag(D)

# Lambda : eigenvalues
lada = np.diag(D) / np.sum(np.diag(D))

# Shannon entropy
Shannon = -np.sum(lada*np.log(lada)) # natural logarithm (log base e)
Effective_rank = np.exp(Shannon)
print('Effective rank =',np.round(Effective_rank,3))

#%%
```

RÉFÉRENCES

[1] G. Henry, « Takning full advantage of microbiological environmental monitoring data ». Nalys, 2017.

[2] M. S. Favero, J. R. Puleo, J. H. Marshall, et G. S. Oxborrow, « Comparative Levels and Types of Microbial Contamination Detected in Industrial Clean Rooms », p. 13, 1996.

[3] G. White, « Pfizer Employees First to Use NCC Cleanroom », avr. 30, 2020. https://www.nashccnews.com/news/2017/03/pfizer-employees-first-to-use-ncc-clean-room/ (consulté le avr. 30, 2020).

[4] EU GMP, « Annex 1 : Manufacture of sterile products ». 2008.

[5] USP, « <1227> VALIDATION OF MICROBIAL RECOVERY FROM PHARMACOPEIAL ARTICLES », p. 3, 2011.

[6] R. A. Caputo et A. Huffman, « Environmental Monitoring: Data Trending Using a Frequency Model », *PDA Journal of Pharmaceutical Science and Technology*, vol. 58, n° 5, p. 9, 2004.

[7] R. Bar, « Charting and Evaluation of Environmental Microbial Monitoring Data », *PDA Journal of Pharmaceutical Science and Technology*, vol. 69, n° 6, p. 743-761, nov. 2015, doi: 10.5731/pdajpst.2015.01079.

[8] D. M. Porter et R. K. Hoffman, « Microbial control in assembly areas needed for spacecraft sterilization », 1965.

[9] H. Yang, « Multivariate Control Chart for Environmental Monitoring », *IVT Network*, p. 9, 2013.

[10] G. Atluri, A. Karpatne, et V. Kumar, « Spatio-Temporal Data Mining: A Survey of Problems and Methods », *ACM Computing Surveys*, vol. 51, n° 4, p. 1-41, août 2018, doi: 10.1145/3161602.

[11] P. Esling et C. Agon, « Time-series data mining », *ACM Computing Surveys*, vol. 45, n° 1, p. 1-34, nov. 2012, doi: 10.1145/2379776.2379788.

[12] M. Vlachos, G. Kollios, et D. Gunopulos, « Discovering similar multidimensional trajectories », dans *Proceedings 18th International Conference on Data Engineering*, San Jose, CA, USA, 2002, p. 673-684, doi: 10.1109/ICDE.2002.994784.

[13] W.-C. Lin et C.-F. Tsai, « Missing value imputation: a review and analysis of the literature (2006–2017) », *Artif Intell Rev*, vol. 53, n° 2, p. 1487-1509, févr. 2020, doi: 10.1007/s10462-019-09709-4.

[14] R. Chotirat Ann, J. Lin, D. Gunopulos, et E. Keogh, « Mining Times Series data », p. 36, 2010.

[15] O. Erdem, E. Ceyhan, et Y. Varli, « A new correlation coefficient for bivariate time-series data », *Physica A: Statistical Mechanics and its Applications*, vol. 414, p. 274-284, nov. 2014, doi: 10.1016/j.physa.2014.07.054.

[16] M. Lu, U. Lall, J. Kawale, S. Liess, et V. Kumar, « Exploring the Predictability of 30-Day Extreme Precipitation Occurrence Using a Global SST–SLP Correlation Network », *J. Climate*, vol. 29, n° 3, p. 1013-1029, nov. 2015, doi: 10.1175/JCLI-D-14-00452.1.

[17] Y. Permanasari, E. H. Harahap, et E. P. Ali, « Speech recognition using Dynamic Time Warping (DTW) », *J. Phys.: Conf. Ser.*, vol. 1366, p. 012091, nov. 2019, doi: 10.1088/1742-6596/1366/1/012091.

[18] G. Atluri, M. Steinbach, K. O. Lim, A. M. Iii, et V. Kumar, « Discovering Groups of Time Series with Similar Behavior in Multiple Small Intervals of Time », dans *Proceedings of the*

2014 SIAM International Conference on Data Mining, avr. 2014, p. 1001-1009, doi: 10.1137/1.9781611973440.114.
[19] W. S. Schlindwein et M. Gibson, Éd., « Pharmaceutical quality by design : A practical approach », dans *Pharmaceutical Quality by Design*, Chichester, UK: John Wiley & Sons, Ltd, 2018, p. i-xxvi.
[20] S. Wold, K. Esbensen, et P. Geladi, « Principal Component Analysis », p. 16, 1987.
[21] R. Tauler, « Multivariate curve resolution applied to second order data », *Chemometrics and Intelligent Laboratory Systems*, vol. 30, n° 1, p. 133-146, nov. 1995, doi: 10.1016/0169-7439(95)00047-X.
[22] A. de Juan, Y. Vander Heyden, R. Tauler, et D. L. Massart, « Assessment of new constraints applied to the alternating least squares method », *Analytica Chimica Acta*, p. 12, 1997.
[23] C. Ruckebusch et L. Blanchet, « Multivariate curve resolution: A review of advanced and tailored applications and challenges », *Analytica Chimica Acta*, vol. 765, p. 28-36, févr. 2013, doi: 10.1016/j.aca.2012.12.028.
[24] O. Roy et M. Vetterli, « The Effective Rank: a Measure of Effective Dimensionality », *European Signal Processing Conference*, vol. 15, p. 6, 2007.
[25] A. Golshan, C. Evans, P. Geary, A. Morrow, Z. Rogers, et M. Maeder, « Multivariate Analytical Insights into Spatial and Temporal Variation in Water Quality of a Major Drinking Water Reservoir », *International journal of Environmental and ecological Engineering*, vol. 12, n° 3, p. 8, 2018.
[26] M. Terrado, D. Barceló, et R. Tauler, « Quality Assessment of the Multivariate Curve Resolution Alternating Least Squares Method for the Investigation of Environmental Pollution Patterns in Surface Water », *Environ. Sci. Technol.*, vol. 43, n° 14, p. 5321-5326, juill. 2009, doi: 10.1021/es803333s.
[27] A. Malik et R. Tauler, « Extension and application of multivariate curve resolution-alternating least squares to four-way quadrilinear data-obtained in the investigation of pollution patterns on Yamuna River, India—A case study », *Analytica Chimica Acta*, vol. 794, p. 20-28, sept. 2013, doi: 10.1016/j.aca.2013.07.047.
[28] M. Alier, M. Felipe, I. Hernández, et R. Tauler, « Trilinearity and component interaction constraints in the multivariate curve resolution investigation of NO and O3 pollution in Barcelona », *Anal Bioanal Chem*, vol. 399, n° 6, p. 2015-2029, févr. 2011, doi: 10.1007/s00216-010-4458-1.
[29] F. B. Lavoie, N. Braidy, et R. Gosselin, « Including noise characteristics in MCR to improve mapping and component extraction from spectral images », *Chemometrics and Intelligent Laboratory Systems*, vol. 153, p. 40-50, avr. 2016, doi: 10.1016/j.chemolab.2016.02.006.
[30] C. Fauteux-Lefebvre, F. Lavoie, et R. Gosselin, « A Hierarchical Multivariate Curve Resolution Methodology To Identify and Map Compounds in Spectral Images », *Anal. Chem.*, vol. 90, n° 21, p. 13118-13125, nov. 2018, doi: 10.1021/acs.analchem.8b04626.
[31] N. Braidy, « Mapping Data with Heavily Overlapped Spectral Features », *Microscopy Society of America*, vol. 23, n° 1, p. 2, 2017.
[32] H. S. Hippert, C. E. Pedreira, et R. C. Souza, « Neural networks for short-term load forecasting: a review and evaluation », *IEEE Transactions on Power Systems*, vol. 16, n° 1, p. 44-55, févr. 2001, doi: 10.1109/59.910780.
[33] D. E. Rumelhart et P. Smolensky, *Schemata and sequential thought processes in PDP models*. 1986.
[34] S. Hochreiter et J. Schmidhuber, « Long Short-Term Memory », *Neural Computation*, vol. 9, n° 8, p. 1735-1780, nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[35] Y. Bengio, P. Simard, et P. Frasconi, « Learning long-term dependencies with gradient descent is difficult », *IEEE Transactions on Neural Networks*, vol. 5, n° 2, p. 157-166, mars 1994, doi: 10.1109/72.279181.
[36] A. Kumar et P. Goyal, « Forecasting of Air Quality Index in Delhi Using Neural Network Based on Principal Component Analysis », *Pure and Applied Geophysics*, vol. 170, n° 4, p. 711-722, avr. 2013, doi: 10.1007/s00024-012-0583-4.
[37] P. Filonov, A. Lavrentyev, et A. Vorontsov, « Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model », *arXiv:1612.06676 [cs, stat]*, déc. 2016, Consulté le: avr. 05, 2019. [En ligne]. Disponible à: http://arxiv.org/abs/1612.06676.
[38] E. Keogh et S. Kasetty, « On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration », p. 23, 2003.
[39] W. A. Fuller, *Introduction to statistical time series*, vol. 428. John Wiley & Sons, 2009.
[40] T. M. J. Fruchterman et E. M. Reingold, « Graph drawing by force-directed placement », *Softw: Pract. Exper.*, vol. 21, n° 11, p. 1129-1164, nov. 1991, doi: 10.1002/spe.4380211102.
[41] D. Hussong et R. E. Madsen, « Analysis of Environmental Microbiology Data from Cleanroom Samples », *Pharmaceutical Technology*, vol. 28, n° SUPPL., p. 10-14, 2004.
[42] R. Tauler *et al.*, « Comparison of the results obtained by four receptor modelling methods in aerosol source apportionment studies », *Atmospheric Environment*, vol. 43, n° 26, p. 3989-3997, août 2009, doi: 10.1016/j.atmosenv.2009.05.018.
[43] M. Dadashi, H. Abdollahi, et R. Tauler, « Maximum Likelihood Principal Component Analysis as initial projection step in Multivariate Curve Resolution analysis of noisy data », *Chemometrics and Intelligent Laboratory Systems*, vol. 118, p. 33-40, août 2012, doi: 10.1016/j.chemolab.2012.07.009.
[44] P. D. Wentzell, « Other Topics in Soft-Modeling: Maximum Likelihood-Based Soft-Modeling Methods », dans *Comprehensive Chemometrics*, Elsevier, 2009, p. 507-558.
[45] A. de Juan, J. Jaumot, et R. Tauler, « Multivariate Curve Resolution (MCR). Solving the mixture analysis problem », *Anal. Methods*, vol. 6, n° 14, p. 4964-4976, 2014, doi: 10.1039/C4AY00571F.
[46] R. Tauler, A. Smilde, et B. Kowalski, « Selectivity, local rank, three-way data analysis and ambiguity in multivariate curve resolution », *Journal of Chemometrics*, vol. 9, n° 1, p. 31-58, janv. 1995, doi: 10.1002/cem.1180090105.
[47] H. Abdollahi et R. Tauler, « Uniqueness and rotation ambiguities in Multivariate Curve Resolution methods », *Chemometrics and Intelligent Laboratory Systems*, vol. 108, n° 2, p. 100-111, oct. 2011, doi: 10.1016/j.chemolab.2011.05.009.
[48] R. Tauler, M. Maeder, et A. de Juan, « Multiset Data Analysis: Extended Multivariate Curve Resolution », dans *Comprehensive Chemometrics*, Elsevier, 2009, p. 473-505.
[49] D. Duan et C. Graff, « UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. » 2019.
[50] NOAA, « Local Climatological Data (LCD) », 2019. https://www.ncdc.noaa.gov/cdo-web/datatools/lcd (consulté le janv. 01, 2020).
[51] X. Sun, S. Kurosu, et H. Shintani, « The Expanded Application of Most Probable Number to the Quantitative Evaluation of Extremely Low Microbial Count », *PDA Journal of Pharmaceutical Science and Technology*, vol. 60, n° 2, p. 13, 2006.
[52] ISO, « ISO 14644-3 : Cleanrooms and associated controlled environments — Part 3: Test methods ». 2019.